

基于 HDF5 的结构网格计算流体动力学程序并行 I/O 技术

杨丽鹏, 车永刚*

(国防科学技术大学 计算机学院, 长沙 410073)

(* 通信作者电子邮箱 ygche@nudt.edu.cn)

摘要:大规模计算流体动力学(CFD)计算对数据 I/O 能力提出了很高需求。层次式文件格式(HDF5)可有效管理大规模科学数据,并对并行 I/O 具有良好的支持。针对结构网格 CFD 并行程序,设计了其数据文件的 HDF5 存储模式,并基于 HDF5 并行 I/O 编程接口实现了其数据文件的并行 I/O,在并行计算机上进行了性能测试与分析。结果表明,在使用 4~32 个进程时,基于 HDF5 并行 I/O 方式的写文件性能比每进程独立写普通文件的方式高 6.9~16.1 倍;基于 HDF5 并行 I/O 方式的读文件性能不及后者,为后者的 20%~70%,但是读文件的时间开销远小于写文件的时间开销,因此对总体性能的影响较小。

关键词:计算流体动力学;结构网格;并行 I/O;层次式文件格式;I/O 性能

中图分类号:TP302.1;TP302.7 **文献标志码:**A

HDF5 based parallel I/O technology for structure grid CFD applications

YANG Lipeng, CHE Yonggang*

(School of Computer, National University of Defense Technology, Changsha Hunan 410073, China)

Abstract: Large scale parallel Computational Fluid Dynamics (CFD) simulation has posed high demand on the data I/O capabilities. Hierarchical Data Format version 5 (HDF5) can effectively manage large scale scientific data and has a good support to parallel I/O. The HDF5 data storage model for a parallel CFD application was designed. The parallel I/O method for the application's data was implemented based on the HDF5 parallel I/O API. Performance experiments were performed on a parallel computer system. The results show that the data writing speed of this HDF5 based parallel I/O method outperforms the parallel I/O methods (i.e. each process writes an ordinary data file independently) by 6.9 to 16.1 times, when 4 to 32 processes are used. The data reading speed of this HDF5 based parallel I/O method is slower than the parallel I/O methods, with a speed of 20% to 70% times of the latter. However, the cost of data reading time is much smaller than the cost of data writing time, hence has minor effect on the total performance.

Key words: Computational Fluid Dynamics (CFD); structure grid; parallel I/O; Hierarchical Data Format version 5 (HDF5); I/O performance

0 引言

计算流体动力学(Computational Fluid Dynamics, CFD)是一门综合了流体力学、计算数学、计算机等学科的交叉学科,它采用数值方法求解流体运动控制方程,获得相应的流动参数,目前已经越来越多地应用在工程中。随着所模拟的几何外形日益复杂,流动机理研究越来越精细,CFD 计算对性能的需求日益增长,成为典型的高性能数值计算应用领域^[1]。CFD 计算对性能需求的增长不仅体现在计算量上,而且也体现在数据规模上。据估计当前流动机理研究(如湍流直接模拟)方面 CFD 使用的网格数达到了数十亿到上百亿量级,在实际工业应用方面 CFD 使用的网格规模也达到了千万量级。CFD 相应的输入文件、中间文件和最终结果文件容量巨大,数据规模从吉字节(GB)级增长到了太字节(TB)级。巨大的数据量对数据 I/O 的性能带来巨大的挑战,如何高效管理和读写数据成为当前 CFD 的重要问题。

并行 I/O 技术^[2]是解决大规模数据输入输出性能需求的主要途径之一^[3]。并行 I/O 就是使一个并行程序的多个

进程同时访问文件中的数据,依据文件组织不同分为两类:每个进程独立访问一个文件和所有进程共同访问单一的共享文件^[4]。学术界对并行 I/O 进行了大量的研究,主要集中在并行 I/O 硬件体系结构、支持并行 I/O 的系统软件以及应用程序层面的并行 I/O 技术。在应用层并行 I/O 方面, Pakalapati^[5]基于扩展的 CFD 通用符号系统(parallel CFD General Notation System, pCGNS)使用基准测试程序(benchmark)测试了 CFD 应用程序的并行 I/O 性能,文中利用 pCGNS 进行数据访问,在两种并行文件系统(网络文件系统 NFS 和并行虚拟文件系统 PVFS)下分别进行并行 I/O 性能的测试并进行了比较,结果证明了 pCGNS 系统的良好性能。Horne 等^[6]也基于 pCGNS 使用两个 benchmark(IOR 和自研的 pCGNS benchmark)对并行 I/O 性能进行测试,文中将 pCGNS benchmark 分为五类,分别从每个进程上的不同分区数对并行 I/O 性能进行测试,并与 IOR benchmark 的测试结果分别进行比较。张娟等^[7]针对划分计算空间到多个计算区域并基于计算区域组织 I/O 的问题,设计了多级极大独立集算法实现计算区域的主进程选择及计算区域数据访问先后

收稿日期:2013-03-28;修回日期:2013-05-09。

基金项目:国家 973 计划项目(G2009CB723803);航天支撑技术基金资助项目(2012-HT-CFKD-19)。

作者简介:杨丽鹏(1986-),女,山东郓城人,硕士研究生,CCF 会员,主要研究方向:并行算法设计、并行 I/O;车永刚(1973-),男,云南嵩明人,研究员,博士,CCF 高级会员,主要研究方向:并行计算、计算机系统性能评测。

顺序指定,分析表明这种方法的 I/O 并行度最高,在并行度最高情况下通信量最小。颜小洋等^[8]三维电磁场粒子模拟并行应用,在分布式存储系统中,采用各节点同时把数据输入输出到本地单独文件中的方式,大幅度减少数据 I/O 的时间。蔡昭权等^[9]针对外汇数据分析应用,利用层次式文件格式(Hierarchical Data Format version 5, HDF5)^[10]的 I/O 性能和数据类型自定义功能进行数据存储,加快数据存储的速度,数据读写性能比 MSSQL 提高了 80 倍左右。冯金花等^[11]以 Delphi 与 Python 的混合编程方式实现了基于 HDF5 文件格式的数据管理系统。

前面提到的 HDF5 是一种层次式数据存储格式,它支持多种科学数据类型,可较好地实现不同应用间的数据共享,并提供了高效的并行 I/O 机制,是解决并行计算中大量数据的快速输入输出的一种重要技术,目前 HDF5 被广泛应用在高性能计算领域。本文针对大规模 CFD 并行计算对数据 I/O 性能的需求,以一个结构网格 CFD 并行计算程序为应用背景,研究采用 HDF5 格式来存储多块结构网格数据,并基于 HDF5 的并行 I/O 机制实现其快速并行读写。在并行计算机系统上针对实际算例进行了性能测试,检验了该技术的有效性,分析了存在的不足,指出了进一步优化的方向。

1 HDF5 数据格式及其并行 I/O

HDF5 是美国国家航空与航天管理局(NASA)、美国国家计算机安全协会与美国三大实验室(IINI、LANL、SNI)合作设计开发,于 1998 年推出的用于存储科学数据的一种新型数据文件格式。HDF5 数据文件采用二进制格式存储科学数据及元数据,提供高效灵活的存储机制,支持磁盘、内存、网络、虚拟设备层存储。相应的 HDF5 数据管理软件支持海量数据的高效存储和访问。

1.1 HDF5 数据格式构成

HDF5 文件有两种基本数据对象:组(groups)和数据集(datasets)。有三种辅助对象:数据类型(datatype)、数据空间(dataspace)和属性(attribute)。HDF5 文件中的数据以图的形式组织,HDF5 文件与根组("/")一一对应,根组中包含一个或多个组,组中又可包含其他组、数据集及元数据(metadata),数据集由多个多维数据元素组成^[12]。HDF5 的命名数据对象(组、数据集、命名的数据类型)可以有 0 个或多个属性,它描述数据对象的结构、大小、数据单位等信息,与所依附的命名数据对象存储在一起。简单的 HDF5 对象间的层次结构如图 1 所示,其中椭圆表示组,矩形表示数据集,圆表示数据集的依附对象。

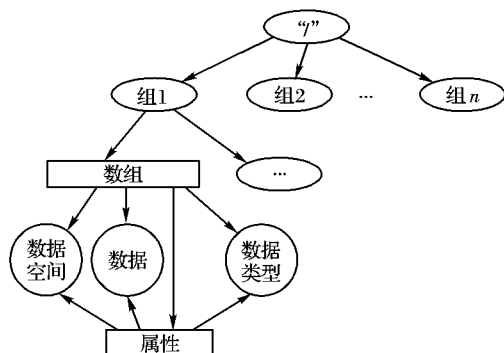


图 1 HDF5 文件层次结构

在 HDF5 的逻辑结构中,整个文件类似于一本多章节的书,每章相当于一个 HDF5 组,每小节相当于一个 HDF5 数据

集,其中每小节都包含一个不同类型的数据内容。层次式是 HDF5 逻辑结构的核心思想,通过层次式的方式,有效地建立文件内对象之间的逻辑包含关系和组织方式。

1.2 HDF5 数据格式特点

HDF5 是一种具有自我描述性、可扩展性、自我组织性的数据存储格式,主要用来存储各种类型的科学数据,适应于多种计算机平台,易于扩展。HDF5 能够创建和存储任意类型的科学数据结构^[13],如图像、多维数组、指针及文本数据等,可根据需要将这些数据类型混合存储在一个 HDF5 文件中,数据对象间可建立数据依赖关系,这就取消了文件系统对单个文件规模的限制。同时,HDF5 利用数据分块、压缩、扩展、外部存储等策略提高数据的存储、访问和管理效率,实现数据在 HDF5 文件间和在应用程序间的共享。

数据描述和数据存在于同一个文件,是 HDF5 的主要特点之一;HDF5 的另一个主要特点是支持文件级的并行读写,且支持分块数据存储,这是 HDF5 为支持高效数据 I/O 而引入的新特征。HDF5 提供了高效灵活的并行 I/O 机制,支持聚合(collective)和独立(independent)两种并行 I/O 方式的数据集读写。若每个数据块以单一 I/O 操作进行读写操作,独立完成数据的传输,则称之为独立并行 I/O;若每个数据块由所有进程共同进行 I/O 操作,则称为聚合并行 I/O。并行 HDF5 实现层次^[14]如图 2 所示,基于 MPI 编程的应用程序调用 HDF5 的并行 I/O 接口函数,HDF5 基于 MPI-IO 来支持并行 I/O。

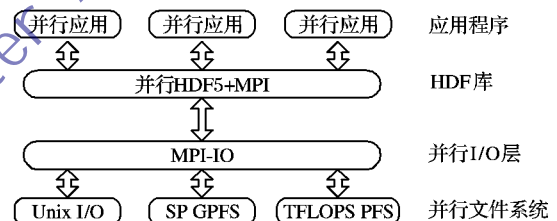


图 2 并行 HDF5 实现层次图

2 结构网格 CFD 并行 I/O

计算网格生成是 CFD 计算的前提,网格按照数据类型和网格单元的几何形状分为结构网格、非结构网格和混合网格。结构网格按照网格点的顺序进行数据存储,可以根据数组下标方便地索引和查找,数据存储开销小,并且由于结构网格具有贴体性,流场计算精度高,在 CFD 计算特别是高精度 CFD 计算中应用较多。由于 HDF5 格式可有效地管理科学数据,并且支持并行 I/O,因此本文研究利用 HDF5 来存储 CFD 中的结构网格数据。

本文结构网格 CFD 程序涉及的数据文件包括网格文件、重启动文件、网格拓扑文件、流场文件等,逻辑上将所有原始文件组织到一个 HDF5 文件中,以方便数据的管理。基于 HDF5 的 CFD 结构网格数据文件结构分为三层:第一层是文件层,相当于根目录;第二层是组,一个组对应一个原始文件;第三层是数据集。数据在 HDF5 中存储的逻辑结构如图 3 所示,根组中包含多个组成员,一个组仅包含一个数据集,以多维数组的形式存储相关信息。数据集具有特有的属性,包括网格块个数、各网格块名称、网格维数、每一维大小、数据类型等信息。

在基于 HDF5 文件存储 CFD 数据文件的基础上,本文重点介绍以聚合方式^[15]实现结构网格数据并行 I/O 的方法。写文件的步骤如图 4 所示,首先,所有进程并行聚合地创建一

个 HDF5 文件和文件中组、数据集对象;然后,根据进程数分割结构网格数据,以便每个进程读写一个网格数据块;最后,向 HDF5 文件的数据集中写入数据。读文件步骤与写文件类似,不再赘述。

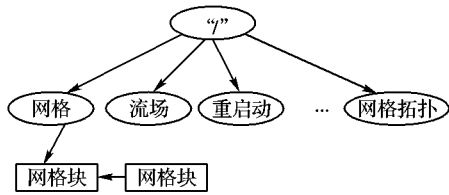


图 3 结构网格数据在 HDF5 中的逻辑结构

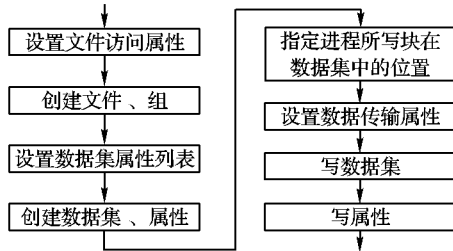


图 4 并行写 HDF5 格式文件的流程

为了实现所有进程并行写单个数据集,首先要存储 MPI_IO 通信器信息到文件访问属性中,然后以此属性创建文件,再依次创建组、数据集和属性。创建数据集时,首先要给数据集分配数据空间(磁盘空间),指定数据类型等,同时也要设置数据集创建属性为分块(chunked)形式,指定数据集中块大小(单个进程负责读写一块数据)。创建属性的方法与创建数据集类似。内存数据空间要与数据集的数据空间匹配,所包含的数据元素个数必须相同,但维数和形状可以不同。使用函数 `h5pset_fapl_mpio_f` 来设置 MPI_IO 形式的文件访问属性,如下所示:

```
h5pset_fapl_mpio_f(fprp_id, comm, info, hdferr)
```

其中:参数 `fprp_id` 是文件属性列表 ID,参数 `comm` 是 MPI 通信器,参数 `info` 是 MPI 消息对象,参数 `hdferr` 是返回码(以下相同变量所表示的意义相同)。参数 `comm` 和 `info` 用于 HDF5 文件的打开。

使用 HDF5 库中的写函数时,要注意内存和数据集空间的选择。所有进程都要参加组、数据集和属性的创建,但写数据集时各进程只负责属于自己的数据块,各进程间以聚合方式进行操作。通过函数 `h5pset_dxpl_mpio_f` 来设置数据传输模式:

```
h5pset_dxpl_mpio_f(dprp_id, data_xfer_mode, hdferr)
```

其中:`dprp_id` 参数表示数据集属性列表 ID,`data_xfer_mode` 参数说明数据传输模式,有两种可选的模式: `H5FD_MPIO_INDEPENDENT_F` 和 `H5FD_MPIO_COLLECTIVE_F`。

本程序中并行 I/O 的关键内容是单数据集分割。在数据分割时,需要设置数据集创建属性为分块存储,在此基础上才能进行整个数据集的分块。对于单个数据集内的并行读写,HDF5 中使用数据分割技术,提供了一个库函数 `h5sselect_hyperslab_f` 用于分块选择:

```
h5sselect_hyperslab_f(space_id, operator, start, count, hdferr, stride, block)
```

其中:`operator` 参数表示分块操作方式,有新建和添加两种方式,其中添加又包含多种不同方式(如逻辑与、逻辑或等)。为了实现单数据集的多进程并行 I/O 读写,每个进程需在多维数据集中指定分块的起始点位置 `start`、跨步大小(元素个数) `stride`、数据单元个数 `count` 和数据单元基准元素个数 `block`。在内存中分配数据空间,空间大小为一个进程负责读

写的数据集块大小,即 HDF5 文件分块存储的块大小。磁盘上数据集所分配的数据空间与内存空间不同,磁盘上分配整个数据集空间,即所有进程所负责写的数据集块的总和。分块方式如图 5 所示,假设有 6 个进程(P0~P5)并行写一个数据集,左半部分 6 个小网格表示各个进程要写入 HDF5 文件的数据集块,右半部分大网格表示整个数据集,每个进程写入数据集中相应的位置。HDF5 库在写数据集时指定数据传输属性为聚合方式,使得每个进程将内存缓冲区中的数据以聚合方式并行写入对应的文件块中,最后完成整个数据集的并行分块写入。

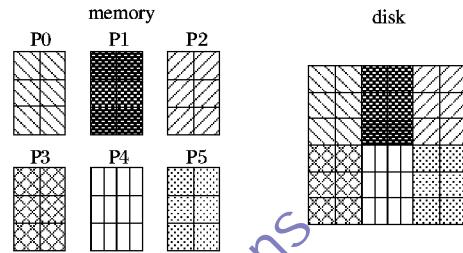


图 5 数据集分块写方式

如果不使用 HDF5 格式,而是使用普通文件格式来存储网格数据,也可以实现并行 I/O,方法是在并行程序并行写文件时,每个进程单独创建一个普通文件,其中包含属于这个进程的网格块数据;读文件时,每个进程各自读取一个文件。本文称这种并行 I/O 方式为 Native 方式。作为对照,本文也实现了基于 Native 方式的结构网格数据并行 I/O 方法。

3 性能测试与评价

3.1 实验环境

在国防科学技术大学高性能计算应用中心一套并行计算机系统上进行性能测试。该系统包含 128 个计算节点,每个节点含 2 个 Intel X5670 处理器(主频 2.93 GHz,6 核,每个核含 32 KB L1 指令 Cache 和 32 KB L1 数据 Cache、256 KB L2 Cache,6 个核共享 12 MB Last Level Cache (LLC))和 48 GB 内存,节点间通过 Infiniband 高速互联,点点通信带宽 40 Gb/s。系统采用 Lustre 全局并行文件系统,共有 6 个数据存储服务器。并行计算环境采用 MPICH2,编译器为 Intel FORTRAN Compiler version 11.1,HDF5 库版本为 `hdf5-1.8.8`。

3.2 测试结果

本文的测试程序是一个三维结构网格 CFD 并行程序^[16],主要考察其网格生成阶段的写网格文件性能和初始化阶段的读网格文件性能。测试所用网格规模分别为 0.75 亿和 2.99 亿,数据类型为单精度浮点数。表 1 给出了两种方式下两种网格规模时的数据文件大小,其中 Native 方式的数据文件大小是全部进程的数据文件大小之和。可以看到,由于在 HDF5 和 Native 方式下数据存储方式均为二进制格式,两种方式下的数据文件大小基本相同。数据均存储在数据存储服务器上,并行 CFD 程序运行在计算节点(下面简称节点)上,通过 Lustre 并行文件系统访问这些数据。

表 1 两种方式下两种网格规模时的数据文件大小 MB

规模	HDF5	Native	规模	HDF5	Native
0.75 亿	859	860	2.99 亿	3430	3428

测试所用进程数分别为 4,8,16 和 32,均在 4 个节点上运行。测试时间均运行 4 遍,取最小值,两种网格规模下的测试结果见表 2。

表 2 两种方式下两种网络规模下的文件读写时间

s

方式	读/写文件	网络规模为 0.75 亿				网络规模为 2.99 亿			
		4 进程	8 进程	16 进程	32 进程	4 进程	8 进程	16 进程	32 进程
HDF5	写	5.643 026	3.811 414	4.360 714	5.015 245	35.096 180	24.396 500	26.876 950	26.708 870
	读	0.654 904	1.370 781	4.906 581	3.995 975	1.064 462	6.942 637	17.961 250	11.466 870
Native	写	95.531 126	48.192 230	31.297 113	21.900 080	513.686 500	192.186 400	118.691 710	87.125 980
	读	3.641 089	1.826 905	0.503 948	0.475 132	15.136 310	7.489 914	3.654 438	1.898 765

根据表 2 数据,图 6 给出了两种网络规模下 HDF5 方式与 Native 方式并行 I/O 写文件的时间对比。从图 6 可以看出,在写文件时,HDF5 方式的并行 IO 性能优于 Native 方式,相同规模下,HDF5 并行 I/O 方式性能为 Native 方式的 6.9 ~ 16.1 倍。在相同网络规模下,进程数越多写文件时间越短,这是由于数据集的每块由一个单独的进程负责写入,进程间不存在额外的通信。进程数越多对数据集分块越小,相对于每个进程来说块空间减少,进而写数据所用的时间变小。

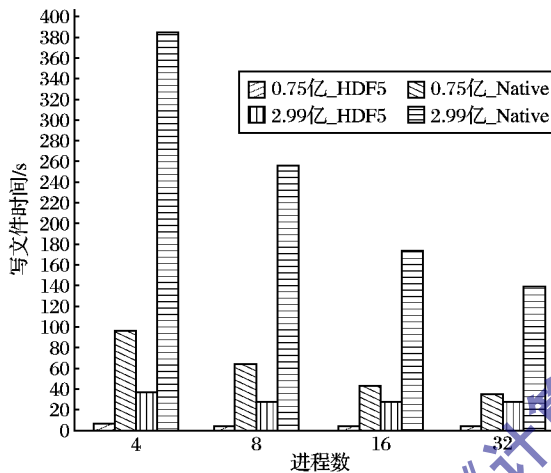


图 6 HDF5 方式与 Native 方式的写文件时间对比

图 7 给出了两种网络规模下 HDF5 方式与 Native 方式的读文件时间对比。从图 7 可以看到,在读文件时,HDF5 方式的性能不如 Native 方式,但是随着网络规模增大,HDF5 方式与 Native 方式的读文件性能差距变小。在进程数增加时,两种方式下的读文件时间都是先增加,之后又有所降低。其原因是由于读文件时间比写文件时间短得多,此时底层并行文件系统和通信系统的带宽利用率均很高,更多的进程会加剧资源的竞争,有可能使性能降低。

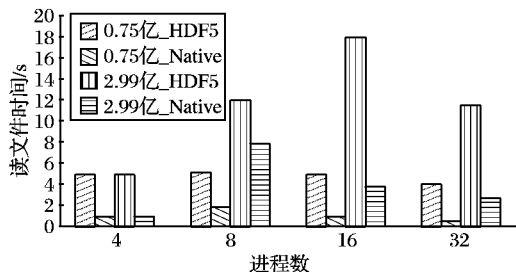


图 7 HDF5 方式与 Native 方式读文件时间对比

将相同数目的进程分配到不同数目的节点上,并行 I/O 性能也会出现差异。图 8~9 给出了网络规模为 0.75 亿时相同进程不同节点数时 HDF5 方式和 Native 方式写文件时间对比。图 10~11 给出了网络规模为 2.99 亿时相同进程不同节点数时 HDF5 方式和 Native 方式写文件时间对比。总体上看,HDF5 方式下,相同进程数时,所用节点数目越多,写文件时间越长,这种趋势在数据规模较大时更明显,因为以 HDF5 聚合方式进行并行 I/O 时,进程间需要同步与通信,而并行计算机系统的节点间同步与通信开销大于节点内同步与通信开

销,节点数越多则总的进程间同步与通信开销越大,从而导致时间增加。然而在 Native 方式下,相同进程数时,通常所用节点数越多写文件时间越短,这是由于 Native 方式各个进程独立地写各自的文件,不存在同步与通信开销,将相同进程分布到更多节点还使得进程间的资源竞争减少,因此时间减少。

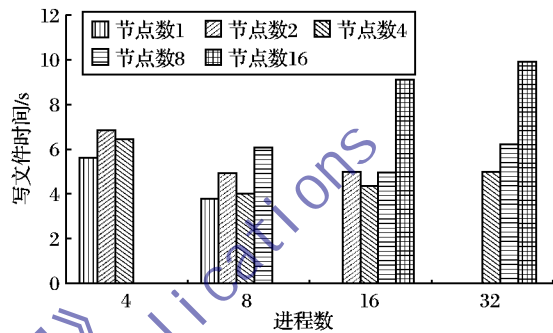


图 8 网络规模为 0.75 亿时 HDF5 写文件时间

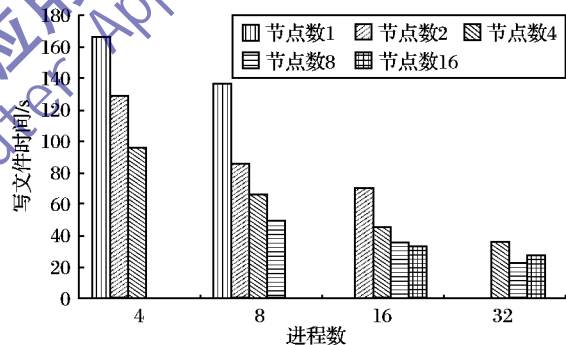


图 9 网络规模为 0.75 亿时 Native 写文件时间

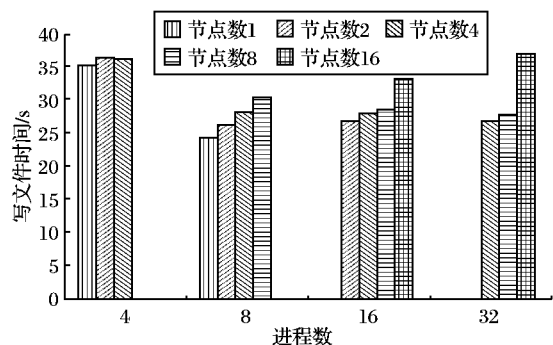


图 10 网络规模为 2.99 亿时 HDF5 写文件时间

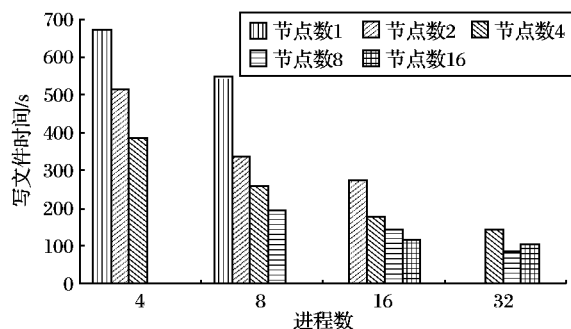


图 11 网络规模为 2.99 亿时 Native 写文件时间

从上述数据还可看到,对相同数据规模,两种方式下读文件的时间均比写文件的时间短得多,故写文件的时间对整体性能的影响更大。另外,在 CFD 计算中,对数据文件的访问方式往往是一次读(读初始网格数据或重启计算文件)、多次写(写中间计算结果文件、重启动文件等),故提高写文件性能具有特别的价值。

4 结语

本文针对结构网格 CFD 并行计算中对数据 I/O 能力的需求,研究了 HDF5 数据存储格式及其并行 I/O 技术,基于 HDF5 实现结构网格 CFD 程序网格数据的并行 I/O。在并行计算机系统上进行了性能测试,结果表明基于 HDF5 并行 I/O 方式的写文件性能比每进程独立写普通文件的并行 I/O 方式高约 6.9 ~ 16.1 倍,基于 HDF5 并行 I/O 方式的读文件性能不及后者,约为后者的 20% ~ 70%,但是读文件的时间开销远小于写文件的时间开销。本文工作表明 HDF5 在结构网格 CFD 程序并行 I/O 中的应用前景。

参考文献:

- [1] 张月玲, 白文. 航空高性能计算和数据处理需求评论[J]. 航空计算技术, 2009, 39(1): 146 - 150.
- [2] DONGARRA J, FOSTER I, FOX G. 并行计算综论[M]. 莫则尧, 陈军, 曹小林, 等译. 北京: 电子工业出版社, 2005.
- [3] 陈志刚, 曾碧卿. HPPHM: 多维数据并行 I/O 存储新策略及其性能研究[J]. 通信学报, 2007, 28(4): 57 - 64.
- [4] 杨伟光. 并行程序中不同 I/O 实现方式的设计与 I/O 性能[D]. 呼和浩特: 内蒙古大学, 2006.
- [5] PAKALAPATI P D. Benchmarking parallel I/O performance for computational fluid dynamics applications[C]// Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit. Washington,

DC: AIAA, 2005.

- [6] HORNE K, BENSON N. An efficient and flexible parallel I/O implementation for the CFD general notation system[R]. Salt Lake City: Utah State University, Center for High Performance Computing, 2009.
- [7] 张娟, 陆林生. 应用层并行 I/O 效率研究[J]. 计算机应用, 2009, 29(1): 9 - 11.
- [8] 颜小洋, 张伟文, 布社辉, 等. 计算与通信重叠和并行 I/O 在粒子模拟中的应用[J]. 计算机应用, 2011, 31(S1): 33 - 39.
- [9] 蔡昭权, 吴文忠, 卢庆武, 等. 利用 HDF5 和 Esper 的高效外汇数据分析系统[J]. 计算机工程与科学, 2011, 33(4): 159 - 163.
- [10] RITTER M. Introduction to HDF 5 and F 5 [D]. Innsbruck: Leopold-Franzens University of Innsbruck, Institute of Computer Science, 2007.
- [11] 冯金花, 陈建政. 基于 HDF5 文件格式的数据管理系统[J]. 计算机与数字工程, 2011, 39(2): 102 - 103.
- [12] 沈卫超, 曹立强, 夏芳, 等. 面向数值模拟数据的 HDF5 性能优化[J]. 计算机研究与发展, 2012, 49(S1): 314 - 318.
- [13] YANG M Q, KOZIOL Q. Parallel HDF5 hints[R]. Urbana-Champaign: The HDF Group, 2010.
- [14] YANG K. Parallel HDF5 introductory tutorial[R]. Urbana-Champaign: The HDF Group, 2008.
- [15] YANG M Q, KOZIOL Q. Using collective IO inside a high performance IO software package — HDF5[R]. Urbana-Champaign: University of Illinois at Urbana Champaign, National Center for Supercomputing Applications, 2010.
- [16] CHEY G, XU C F, WU J P, et al. Optimization and evaluation of a low mach number flow simulator on tianhe-1A supercomputer [C]// Proceedings of the 23th International Conference on Parallel Computational Fluid Dynamics. Barcelona: IEC, 2011.

(上接第 2418 页)

3 结语

针对典型的一维双曲守恒律方程(组)的数值求解问题,本文构造了四阶 CWENO 型熵相容算法,并将新算法应用于若干问题的数值求解当中。通过对所得结果的比较与分析,所得结论如下:

- 1) 新算法所得数值结果正确;
- 2) 新算法不仅可以处理 Roe 型格式不能处理的静态激波问题,而且保留了 Roe 型格式能准确捕捉激波的特点;
- 3) 新算法可以避免强稀疏波问题中的膨胀激波产生;
- 4) 新算法比三阶 CWENO 型熵相容算法分辨率高。

新算法在空间方向离散时引入了四阶 CWENO 重构,时间方向采用半离散方式,该算法易于编程实现。

参考文献:

- [1] OSHER S. Riemann solvers, the entropy condition, and difference approximations [J]. SIAM Journal on Numerical Analysis, 1984, 21(2): 217 - 235.
- [2] TADMOR E. Numerical viscosity and the entropy condition for conservative difference schemes [J]. Mathematics of Computation, 1984, 43(168): 369 - 381.
- [3] TADMOR E. The numerical viscosity of entropy stable schemes for systems of conservation laws [J]. Mathematics of Computation, 1987, 49(179): 91 - 103.
- [4] MOHAMMED A-N, ISMAIL F. Study of an entropy-consistent Navier-Stokes flux[J]. International Journal of Computational Fluid Dynamics, 2013, 27(1): 1 - 14.
- [5] ISMAIL F, ROE P-L. Affordable, entropy-consistent Euler flux

functions II: entropy production at shocks [J]. Journal of Computational Physics, 2009, 228(15): 5410 - 5436.

- [6] ISMAIL F. Towards a reliable prediction of shocks in hyperbolic flow: resolving carbuncles with entropy and vorticity control [D]. Ann Arbor, Michigan: University of Michigan, 2006.
- [7] FJORDHOLM U-S, MISHRA S, TADMOR E. Entropy stable ENO scheme [R]. Zürich: Eidgenössische Technische Hochschule Zürich, 2011.
- [8] FJORDHOLM U-S, MISHRA S, TADMOR E. Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws [J]. SIAM Journal on Numerical Analysis, 2012, 50(2): 544 - 573.
- [9] 郑素佩, 封建湖, 刘彩侠. 二维双曲守恒律标量方程的三阶 CWENO-型熵相容算法[J]. 计算机应用, 2012, 32(10): 2745 - 2747.
- [10] LEVY D, PUPPO G, RUSSO G. A third order central WENO scheme for 2D conservation laws [J]. Applied Numerical Mathematics, 2000, 33(1/2/3/4): 407 - 414.
- [11] LEVY D, PUPPO G, RUSSO G. A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws [J]. SIAM Journal on Scientific Computing, 2002, 24(2): 480 - 506.
- [12] KURGANOV A, NOELLE S, PETROVA G. Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations [J]. SIAM Journal on Scientific Computing, 2001, 23(3): 707 - 740.
- [13] 罗力. 求解双曲守恒律方程的熵稳定格式研究[D]. 西安: 长安大学, 2011.