

边缘海静力数值预报模式并行算法研究

逢仁波^{1,2} 张云泉³ 谭光明³ 徐建良¹ 贾海鹏³ 解庆春⁴

(中国海洋大学信息科学与工程学院 青岛 266100)¹

(国家海洋环境预报中心网络与计算机部 北京 100081)²

(中国科学院计算技术研究所计算机体系结构国家重点实验室 北京 100190)³

(中国传媒大学高性能计算中心 北京 100024)⁴

摘要 边缘海静力数值模式是国内针对边缘海特点自主开发的数值预报模式,但该模式因物理求解方程较多且采用不宜并行化的 SOR 求解算法而程序计算时间过长。针对上述问题,提出基于三维网格和海洋模式特点的 SOR 并行求解算法,该算法在保留三维网格数据间依赖关系的同时,有效解决了 SOR 迭代算法难以并行化的问题。同时,引入通信避免算法,采用 MPI 非阻塞通信方式,细分计算和通信过程,利用计算有效隐藏通信开销,提高了并行程序效率。实验结果表明,并行后的边缘海静力数值模式程序的性能相对串行程序提升了 60.71 倍,3 天(25920 计算时间步)预报结果的均方根误差低于 0.001,满足海洋数值预报的时效性和精度要求。

关键词 数值模式,预报,并行,非阻塞,SOR

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.1.003

Parallelization of Hydrostatic Numerical Forecasting Model of Marginal Sea

PANG Ren-bo^{1,2} ZHANG Yun-quan³ TAN Guang-ming³ XU Jian-liang¹ JIA Hai-peng³ XIE Qing-chun⁴

(College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China)¹

(Department of Computer and Network, National Marine Environmental Forecasting Center, Beijing 100081, China)²

(State Key Laboratory of Computer Architecture, ICT, CAS, Beijing 100190, China)³

(High Performance Computing Center, Communication University of China, Beijing 100024, China)⁴

Abstract The hydrostatic numerical forecasting model of marginal sea is the numerical forecasting model developed independently in China according to the characteristics of the marginal sea. There are many physical equations in the model, and some equations are not appropriate for parallelism like SOR, so it takes a long time to run the program. To solve these problems, the parallel SOR method, which is based on 3D computing grids and characters of ocean forecasting models, is used to solve the parallel problem and keep dependencies between data in 3D computing grids. The methods are also used to improve the efficiency of the parallel program, such as using MPI non-blocking communication, cutting the process of computing and communication to more steps, and overlapping the steps of communication with computing. The experiments show that the performance of the parallel hydrostatic numerical forecasting model increases 60.71 times compared to the serial program, and root mean square error of three-day forecasting results (25920 time steps) is less than 0.001, which meets the numerical ocean forecasting requirements of timeliness and accuracy.

Keywords Numerical model, Forecasting, Parallelization, Non-blocking, SOR

1 引言

海洋数值模式是当前海洋预报业务的基础,对预报精度和时效改进起着关键作用。针对中国边缘海特点,国内自主开发了边缘海静力数值预报模式(Hydrostatic Numerical Forecasting Model of Marginal Sea, HYNFOM),以提供流

速、温度、压强、盐度和密度等预报结果。因该模式包含复杂的物理计算方程和算法,导致模式计算时间过长,例如计算未来 3 天的预报结果, HYNFOM 串行程序需要 3.3 个小时,无法满足数值预报时效性要求,实现模式的程序并行化是必然趋势。

在程序并行化开发中,数据分区、通信优化、负载均衡等

到稿日期:2015-03-09 返修日期:2015-04-12 本文受海洋公益性行业科研专项:海洋渔业安全环境保障服务系统关键技术研究及示范应用(201205006),国家自然科学基金面上项目(61272136),国家自然科学基金重点基金(61133005),国家自然科学基金创新群体成员(60921002),中国科学院计算技术研究所创新课题(20136030),中国科学院计算技术研究所计算机体系结构国家重点实验室创新课题(CARCH301)资助。

逢仁波(1982-),男,博士生,主要研究方向为大规模并行计算和优化,E-mail: pangrb@nmefc.gov.cn;张云泉(1973-),男,研究员,主要研究方向为大型并行数值软件、并行程序设计和性能评价、并行计算模型、海量数据并行处理等;谭光明(1980-),男,研究员,主要研究方向为高性能计算;徐建良(1969-),男,研究员,主要研究方向为计算复杂性理论、计算机软件与理论、人工智能;贾海鹏(1983-),男,博士后,主要研究方向为大规模并行算法与软件、异构众核编程方法及优化技术;解庆春(1982-),男,助理研究员,主要研究方向为计算机软件与理论、高性能计算。

方法被广泛采用^[1-6],但由于 HYNFOM 包含隐式迭代方程等求解算法,除通用的并行优化方法外,还需要根据模式不同的算法进行专门并行化设计。例如,SOR 迭代求解算法在 HYNFOM 中占用 17% 的计算量,且不适宜并行化。

目前,针对 SOR 算法的并行化方法主要有红黑排序^[7-9]、多色排序^[10,11]、多分算法^[12-15]等,其中红黑排序和多色排序适用于大型稀疏矩阵,多分算法适用于大型稠密矩阵^[16,17]。多分算法^[18]可以通过矩阵分裂技术实现可并行的松弛迭代算法,但与 SOR 迭代方法相比,其收敛的条件比较严格,收敛速度目前也不太令人满意。HYNFOM 需要基于三维坐标系对 SOR 迭代算法进行求解,且网格间的依赖关系需要符合物理海洋模式特点。

在 SOR 并行算法实现方面^[19,20],主要采用数据分解的方法,将计算任务中计算热点的数据分解成彼此相互独立的部分,然后将这些数据分配给不同的处理器同时计算,以达到提高计算速度的目的。在数据分解方法中,并行计算的效率取决于分解后的数据彼此间独立的程度以及各处理器进行必需的同步操作的频率。

本文的主要贡献为:

- 1) 提出基于三维网格和符合海洋模式特点的 SOR 并行算法,其具有较高的并行度和较低算法复杂度;
- 2) 引入通信避免算法,利用计算有效隐藏通信开销,在实际并行测试中获得了较高的加速性能;
- 3) 实现 SOR 并行算法的同时,保留数据间依赖关系,实验结果表明并行和串行 SOR 求解结果间的误差较小。

2 HYNFOM 程序分析

HYNFOM 主要利用海水动量方程、连续方程、温盐方程、状态方程、湍封闭方程,以及相关的边界条件对流速、水位、盐度和水温等进行求解,物理求解过程较为复杂,其中的主要方程如下所示。

首先,从输入文件中读取经度、纬度、水深、科氏力等参数值;其次,通过水平动量式(1)、式(2)分别求取 x 、 y 坐标系下的水平运动速度 u 、 v 。其中, t 为时间, f 为科氏力参数, g 为重力加速度, D 为总水深, p 为海表大气压强, DIF 为动压方程, K_M 为垂向湍粘性系数, q 为压强中动压值, σ 表示 sigma 坐标值。

$$\frac{\partial uD}{\partial t} = -Adv(uD) + fvD - gD \frac{\partial \eta}{\partial x} - \frac{D}{\rho_0} \frac{\partial p_a}{\partial x} - \frac{gD^2}{\rho_0} \int_{\sigma}^0 \left[\frac{\partial p}{\partial x} - \frac{\sigma'}{D} \frac{\partial D}{\partial x} \frac{\partial \rho}{\partial \sigma} \right] d\sigma' - \frac{D}{\rho_0} \left[\frac{\partial q}{\partial x} + \frac{\partial \sigma}{\partial x} \frac{\partial q}{\partial \sigma} \right] + DIF_u + \frac{\partial}{\partial \sigma} \left(\frac{K_M}{D} \frac{\partial u}{\partial \sigma} \right) \quad (1)$$

$$\frac{\partial vD}{\partial t} = -Adv(vD) + f\mu D - gD \frac{\partial \eta}{\partial y} - \frac{D}{\rho_0} \frac{\partial p_a}{\partial y} - \frac{gD^2}{\rho_0} \int_{\sigma}^0 \left[\frac{\partial p}{\partial y} - \frac{\sigma'}{D} \frac{\partial D}{\partial y} \frac{\partial \rho}{\partial \sigma} \right] d\sigma' - \frac{D}{\rho_0} \left[\frac{\partial q}{\partial y} + \frac{\partial \sigma}{\partial y} \frac{\partial q}{\partial \sigma} \right] + DIF_v + \frac{\partial}{\partial \sigma} \left(\frac{K_M}{D} \frac{\partial v}{\partial \sigma} \right) \quad (2)$$

利用准静力平衡关系,在垂向动量方程中去掉重力和静压垂向梯度力,根据已知的水平流速 u 、 v ,对垂向流速 w 进行求解,如式(3)所示。

$$\frac{\partial wD}{\partial t} = -Adv(wD) - \frac{1}{\rho_0} \frac{\partial q}{\partial \sigma} + DIF_w + \frac{\partial}{\partial \sigma} \left(\frac{K_M}{D} \frac{\partial w}{\partial \sigma} \right) - 2\Omega \cos \varphi u \quad (3)$$

利用三维体积连续方程,根据已知的流速 u 、 v 、 w ,对水位

η 进行求解,如式(4)所示。

$$\frac{\partial uD}{\partial x} + \frac{\partial vD}{\partial y} + \frac{\partial w}{\partial \sigma} + \frac{\partial \eta}{\partial t} = 0 \quad (4)$$

利用海水盐度方程,根据已知的流速 μ 以及由水位 η 求解的总水深 D ,求解盐度 S ,如式(5)所示。

$$\frac{\partial SD}{\partial t} = -Adv(SD) + DIF_S + \frac{\partial}{\partial \sigma} \left(\frac{K_H}{D} \frac{\partial S}{\partial \sigma} \right) \quad (5)$$

根据水温求解方程,利用已知的流速和 μ 以及总水深 D ,对温度 T 进行求解,如式(6)所示,其中 SR_T 是水体所吸收的太阳短波辐射值。

$$\frac{\partial TD}{\partial t} = -Adv(TD) + DIF_T + \frac{\partial}{\partial \sigma} \left(\frac{K_H}{D} \frac{\partial T}{\partial \sigma} \right) + SR_T \quad (6)$$

3 并行算法设计

3.1 网格划分

边缘海数值模式采用三维网格,分别用 x 、 y 、 z 坐标系表示,假如每个计算进程上的网格大小为 (N_x, N_y, N_k) ,划分方式如图 1 所示。根据海洋预报模式特点, z 方向的网格数相对于 x 、 y 方向较少,并行程序仅针对 x 、 y 方向进行网格划分。因此,程序并行度取决于 $N_x \times N_y$,因 N_z 值较小,故该划分方法依然具有较高的并行度。

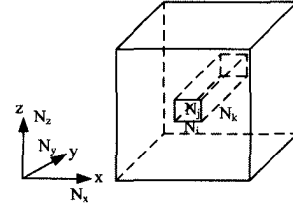


图 1 三维网格划分

网格划分算法如式(7)所示,其中 N_x 、 N_y 、 N_z 分别为 x 、 y 、 z 方向的最大网格数, p 为计算进程总数, N_i 、 N_j 、 N_k 分别为进程在 x 、 y 、 z 方向分配的网格大小。当 N_x 或 N_y 不能被 p 整除时,余数将平均分配到前 $N_x \% p$ 或 $N_y \% p$ 个进程中。

$$\begin{aligned} N_i &= \frac{N_x}{p} + 1 (0 \leq i < N_x \% p), N_i = \frac{N_x}{p} (N_x \% p \leq i < p) \\ N_j &= \frac{N_y}{p} + 1 (0 \leq j < N_y \% p), N_j = \frac{N_y}{p} (N_y \% p \leq j < p) \\ N_k &= N_z (0 \leq k < p) \end{aligned} \quad (7)$$

3.2 SOR 算法并行化

3.2.1 SOR 算法介绍

SOR 算法主要用于对线性方程组进行求解,线性方程组如式(8)所示。其中, a_{11}, \dots, a_{m1} 和 b_1, \dots, b_n 为已知值,通过循环迭代算法对 x_1, \dots, x_n 进行求解。

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (8)$$

通过等式变换,以及引入超松弛因子 ω ,式(8)可转化为式(9),其中 s 为时间步。当 $\omega > 1$ 时,该公式为超松弛迭代法,即 SOR 算法。SOR 算法因迭代求解有先后关系,利用已知的前 $i-1$ 个第 $s+1$ 时间步求解的 x 值和后 $n-i$ 个第 s 时间步求解的 x 值求解 $x_i^{(s+1)}$,逐步循环求解,直至 $x_i^{(s+1)}$ 与 x_i^s 之差在期待误差范围内,从而完成 SOR 算法求解。

$$x_i^{(s+1)} = x_i^s + \frac{\omega}{a_{ij}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(s+1)} - \sum_{j=i+1}^n a_{ij}x_j^s \right) \quad (9)$$

3.2.2 SOR 算法并行化

SOR 主要应用于三维体积连续方程中,利用式(4)对水位 η 进行求解。SOR 串行求解算法较为简单,在单个时间步内,从(1,1,1)坐标开始依次沿 x 、 y 、 z 坐标系进行顺序求解。因此,该算法不具有并行性,但该算法在程序中的计算量占 17%,若不进行并行化,则理论最大加速比为 5.88。

根据海水运动的物理过程,假设网格点在 x 、 y 、 z 的坐标依次为 (N_i, N_j, N_k) ,则网格点 $(N_i, N_j, 2)$ 与相邻的 4 个灰色格点 $((N_i-1, N_j, 2), (N_i, N_j-1, 2), (N_i, N_j+1, 2), (N_i+1, N_j, 2))$ 以及下层格点 $(N_i, N_j, 1)$ 和上层格点 $(N_i, N_j, 3)$ 关系最密切,如图 2 所示。

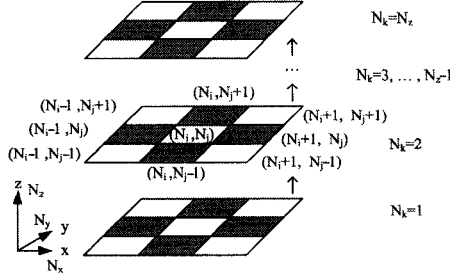


图 2 SOR 并行算法中的网格划分

在并行算法设计中,先对 z 坐标系进行隔层划分,再对 x 、 y 坐标系进行二分分解,从而将单个时间步计算分成 4 部分。

第 1 步 按照式(10),针对 $i \in (1, 2, \dots, N_x), j \in (2 - N_j \% 2, 4 - N_j \% 2, \dots, N_y), k \in (1, 3, \dots, N_k)$ 的网格点进行计算,其中, i 为 x 坐标范围内任意值; k 为 z 坐标范围内奇数值; j 与起始点有关,若为奇数行,则 j 为 y 坐标范围内偶数值,若为偶数行,则 j 为 y 坐标范围内奇数值,如图 2 中 N_k 为奇数的白色网格点。

第 2 步 按照式(10),针对 $i \in (1, 2, \dots, N_x), j \in (2 - N_j \% 2, 4 - N_j \% 2 + 2, \dots, N_y), k \in (2, 4, \dots, N_k)$ 的网格点进行计算, i, j 与第 1 步一致, k 与第 1 步不同,第 2 步中 k 为 z 坐标范围内偶数值,如图 2 中 N_k 为偶数的白色网格点。

第 3 步 按照式(10),针对 $i \in (1, 2, \dots, N_x), j \in (1 + N_j \% 2, 3 + N_j \% 2, \dots, N_y), k \in (1, 3, \dots, N_k)$ 的网格点进行计算,其中, i, k 与第 1 步相同;若 j 的起始坐标值为奇数,则 j 为 y 坐标范围内偶数值;反之,则 j 为 y 坐标范围内奇数值,如图 2 中 N_k 为奇数的灰色网格点。

第 4 步 按照式(10),针对 $i \in (1, 2, \dots, N_x), j \in (1 + N_j \% 2, 3 + N_j \% 2, \dots, N_y), k \in (2, 4, \dots, N_k)$ 的网格点进行计算,其中, i, j 与第 3 步相同, k 与第 3 步不同, k 为 z 坐标范围内偶数值,如图 2 中 N_k 为偶数的灰色网格点。

根据此步骤,利用数据分解的并行方法,依此循环求解,直至 $x_{(i,j,k)}^{(s+1)}$ 与 $x_{(i,j,k)}^s$ 的差值满足精度要求,完成 SOR 算法并行求解。

$$x_{(i,j,k)}^{(s+1)} = x_{(i,j,k)}^s + \frac{\omega}{a_{ijk}} \left(b_i - \sum_{\substack{1 \leq i' < N_x \\ 1 \leq j' \leq N_y \\ 1 \leq k' < N_z}} a_{ijk'} x_{(i',j',k)}^{(s+1)} - \sum_{\substack{N_i \leq i' < N_x \\ 1 \leq j' \leq N_y \\ N_k \leq k' < N_z}} a_{ijk'} x_{(i',j',k)}^{(s+1)} \right) \quad (10)$$

3.2.3 SOR 并行算法复杂度分析

SOR 并行算法在三维网格中的计算复杂度与基于二维网格的红黑排序算法一样,但单次迭代中数据划分比红黑排序更细,是红黑排序算法的 2 倍,从而保留了海洋模式中网格间的特殊依赖关系。

SOR 并行算法的总时间为 T_p ,串行算法的总时间为 T_s ,单次通信时间为 T_u ,单次迭代计算时间为 T_c ,并行变量初始化等并行开销时间为 T_i ,计算进程数为 p ,加速比为 α ,则加速比如式(11)所示。其中, T_i 相对于 T_u 和 T_c 非常少,对性能的影响几乎可以忽略不计;在单次迭代中,需要进行 4 次通信,因此缩短通信时间对加速比具有较大的影响。

$$\alpha = \frac{T_s}{T_p} = \frac{T_c}{T_i + 4T_u + \frac{T_c}{p}} \quad (11)$$

3.2.4 通信避免算法

为了减少通信时间,采用 MPI 非阻塞式通信方式,在循环判断通信结果的过程中加入计算,实现计算和通信重叠,避免通信开销,算法如下所示。

1. MPI_Irecv 接送边界数据;
2. For $i \in N_i, j \in N_j, k \in N_k // N_i, N_j, N_k$ 为每个进程在 x, y, z 坐标系下的网格范围
3. If (MPI_Test(接收是否完成))
4. 计算边界数据; //接收完成
5. Else
6. 计算内部数据; //未接收完成
7. MPI_Isend 发送当前迭代内的边界数据,然后进入下一步迭代计算。

4 性能剖析

4.1 实验环境搭建

实验基于 3 种不同的硬件平台,分别为曙光多核单节点、神威至强集群和神威安腾集群。曙光多核单节点采用英特尔 X7550 CPU(128 核和 2GHz 主频)和 512GB 内存。神威至强集群中每个节点采用英特尔至强 X7350 CPU(16 核和 2.93GHz 主频)和 64GB 内存,节点间通信采用 20Gb/s 的 Infiniband 网络。神威安腾集群中每个节点采用英特尔安腾 9150M CPU(8 核和 1.6GHz 主频)和 16GB 内存,节点间通信采用 20Gb/s 的 Infiniband 网络。

4.2 性能分析

边缘海数值模式并程序在 3 种硬件平台上的加速比曲线如图 3 所示,运行时间如图 4 所示。

在性能测试中,每颗 CPU 核只运行 1 个进程。在神威至强集群上,当计算进程数为 512 时,并行程序运行速度最快,加速比达到 60.71。由于 SOR 迭代求解算法在串行中占用 17% 的计算量,若该算法不进行并行化,并行程序理论最大加速比为 5.88。因此,通过测试程序整体并行效率,依然能够证明 SOR 并行算法具有较好的加速效果。

总体上,随着计算进程数增多,并行程序运行时间变短。当进程数超过 64 个时,加速比增长趋于缓慢,主要原因在于测试采用实际海洋预报业务化应用中常用的网格大小,即 $255 \times 300 \times 11$,网格数量和测试数据相对较少。因为数值模式属于日常业务化运行程序,采用实际网格大小具有较强的实用指导意义。

串程序在曙光单节点上的运行时间最短(9108s),其次是至强集群的单节点(11024s),最后是安腾集群单节点(11957s)。主要原因在于串程序只能在单颗 CPU 上运行,计算节点总体性能对程序影响较少,而 CPU 性能对程序性能影响较大。因为曙光单节点采用 X7550 CPU,至强节点采用 X7350 CPU,安腾节点采用安腾 9150MB CPU,所以串程序在曙光单节点上运行效率高于神威至强节点,神威至强节点高于神威安腾节点。

神威安腾集群的加速比效果优于神威至强,而神威至强集群加速比效果优于曙光单节点。主要原因在于神威安腾集群采用小型机架构,硬件配置合理, Linpack 测试效率达到 85%;而神威至强集群采用 X86 服务器架构, Linpack 测试效率为 78%,故神威安腾集群的加速比较高。

曙光单节点在进程数不超过 16 时,并行程序运算速度较快,计算时间较短。主要原因在于数据在节点内交换,内存和 IO 带宽满足所有 CPU 运算需求;但随着 CPU 核数增多,单节点优势下降,内存和 IO 带宽开始出现瓶颈,导致单节点运算速度慢于神威安腾和至强集群。由于曙光单节点最大核数为 128 颗,因此未进行更高进程数的并行程序性能测试。

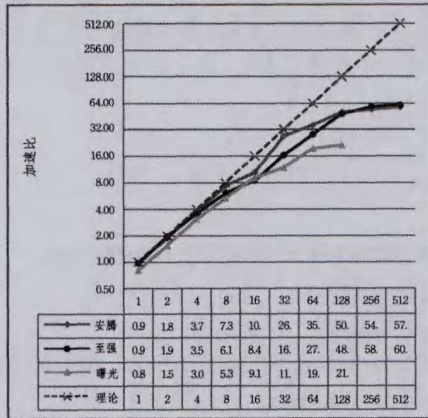


图 3 并行程序加速比曲线

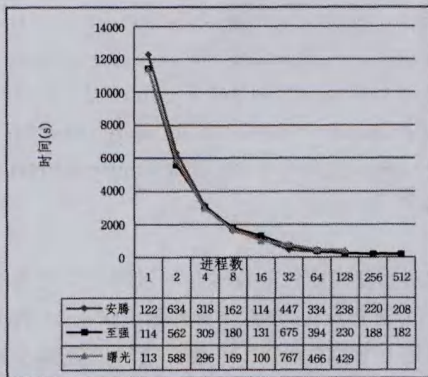


图 4 并行程序运行时间曲线

4.3 误差分析

边缘海串行与并行 SOR 算法的迭代顺序不同,导致两者之间存在一定误差值。利用串行和并行程序分别针对未来 3 天(25920 时间步)的计算结果进行分析,统计出所有计算格点的均方根误差值,如表 1 所列。

表 1 串行和并行程序的均方根误差

进程总数	水深 (η)	水平流速 (u)	垂向流速 (v)	盐度 (s)	密度 (p)
1	0.000672	0.000259	0.000427	0.000662	0.000664
2	0.000674	0.000262	0.000430	0.000682	0.000676
4	0.000677	0.000263	0.000431	0.000624	0.000625
8	0.000673	0.000261	0.000429	0.000494	0.000492
16	0.000670	0.000261	0.000428	0.000790	0.000786
32	0.000675	0.000262	0.000431	0.000497	0.000496
64	0.000674	0.000260	0.000428	0.000828	0.000827
128	0.000664	0.000261	0.000426	0.000623	0.000625
256	0.000674	0.000263	0.000429	0.000488	0.000486
平均误差	0.000672	0.000259	0.0004288	0.000632	0.0006308

根据海洋预报精度实际需求,3 天预报结果的均方根误

差不能超过 0.01,根据统计结果,并程序序和串程序序的均方根误差均在 0.001 以内。由于计算进程总数不同,导致每个进程上分配的网格数量不一致,在迭代求解过程中,也会存在细微的差别,故计算进程数量不同,误差值也不一致,但所有误差值均在精度允许范围内。

根据程序计算结果中的水平流速和垂向流速,绘制流速预报图,串程序序如图 5 所示,并程序序如图 6 所示。从图中可以看出串行流速预报和并行流速预报显示结果一致。

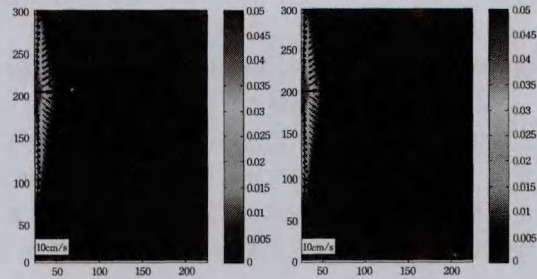


图 5 串程序的流速计算结果 图 6 并行程序的流速计算结果

根据串行程序运行结果中的盐度 S ,绘制盐度分布图,如图 7 所示;并行程序的盐度计算结果如图 8 所示。可以看出串行程序计算结果和并行程序计算结果显示一致。

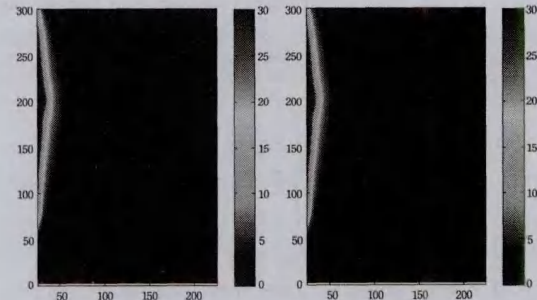


图 7 串程序的盐度计算结果 图 8 并行程序的盐度计算结果

结束语 本文实现了国内自主开发的边缘海数值模式并行化,在三维网格中实现了 SOR 算法并行求解,取得了较好的加速比效果,大大提高了模式预报的时效性。在误差方面,串行和并行程序针对 3 天预报结果的均方根误差均在 0.001 内,满足预报产品的精度要求。

在下一步工作中,将对网格划分方式进行改进,由目前的三维固定边界划分方式改成动态变边界划分方式,实现动态负载均衡,进一步提高效率。在误差控制方面,尝试其他并行化算法,以进一步缩小误差值,为该海洋预报模式业务化应用奠定基础。

参考文献

- [1] Hueske F, Markl V. Optimization of Massively Parallel Data Flows [M]//Large-Scale Data Analytics, 2014; 41-74
- [2] Wu Xing, Liu Yan. Optimization of Load Adaptive Distributed Stream Processing Services [C]// Anchorage; Services Computing, 2014; 504-511
- [3] Ren Xiao-guang, Zhou Wen-hao, Chen Juan. Collective Communication Optimization for Solving Linear Algebraic Equations [J]. Advanced Materials Research, 2014, 989-994: 4934-4939
- [4] Liu Li, Li Rui-zhe, Yang Guang-wen, et al. Improving Parallel Performance of a Finite-Difference AGCM on Modern High-Performance Computers [J]. Oceanic Technol., 2014 (31): 2157-2168

(下转第 29 页)

- [6] Han G W. Research on sequential pattern algorithm for data streams based on prefix sequence tree[D]. Qinhuangdao: Yanshan University, 2013 (in Chinese)
韩高伟. 基于前缀序列树的数据流序列模式算法研究[D]. 秦皇岛: 燕山大学, 2013
- [7] Saputra D, Rambli D R A, Foong O M. Sequential Pattern Mining using PrefixSpan with Pseudoprojection and Separator Database[C]//International Symposium on Information Technology, 2008. Kuala Lumpur, Malaysia, 2008: 1-7
- [8] Dai Y X, Lian Y F, Wang H. System security and intrusion detection[M]. Beijing: Tsinghua University Press, 2002 (in Chinese)
戴英霞, 连一峰, 王航. 系统安全与入侵检测[M]. 北京: 清华大学出版社, 2002
- [9] Guo L, Xiang X, Shi Y C. Use Web usage mining to assist online E-Learning assessment[C]//IEEE International Conference on Advanced Learning Technologies, 2004: 912-913
- [10] Xing G, Shen J. Efficient data mining for web navigation patterns [J]. Information and Software Technology, 2004, 46(1): 55-63
- [11] Xu H Q, Wang Y C. The path Webpage pre fetching model based on analyzing user access[J]. Journal of Software, 2003, 14(6): 1142-1147 (in Chinese)
许欢庆, 王永成. 基于用户访问路径分析的网页预取模型[J]. 软件学报, 2003, 14(6): 1142-1147
- [12] Yang Z L, Wang Y T, Kitsuregawa M M. An Effective system for mining web log[C]// Proceedings of the 8th Asia-Pacific Web Conference on Frontiers of WWW Research and Development (APWeb'06). Harbin, China, 2006: 40-52
- [13] Mabroukeh N R, Ezeife C I. A Taxonomy of Sequential Pattern Mining Algorithms[J]. Journal of ACM Computing Surveys, 2010, 43(1): 1-41
- [14] Hu Y H, Wu F, Liao Y J. An efficient tree-based algorithm for mining sequential patterns with multiple minimum supports[J]. Journal of Systems and Software, 2013, 86(5): 1224-1238
- [15] Hu Y H, Huang C K T, Kao Y H. Knowledge discovery of weighted RFM sequential patterns from customer sequence databases[J]. Journal of Systems and Software, 2013, 86(3): 779-788
- [16] Salehi M, IKamalabadi N. Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree[J]. Knowledge-Based Systems, 2013, 48: 57-69
- [17] Hu Y H, Tsai C F, Tai C T, et al. A novel approach for mining cyclically repeated patterns with multiple minimum supports [J]. Applied Soft Computing, 2015, 28: 90-99
- [18] Zhang B B, Lin C W, Gan W S, et al. Maintaining the discovered sequential patterns for sequence insertion in dynamic databases [J]. Engineering Applications of Artificial Intelligence, 2014, 35(10): 131-142

(上接第 17 页)

- [5] Yan Chao-kun, Hu Zhi-gang, Li Xi, et al. Reliability-Cost Optimization Scheduling Model and Algorithm in Grid [J]. Computer Science, 2013, 40(3): 136-141 (in Chinese)
阎朝坤, 胡志刚, 李玺, 等. 面向可靠性-费用优化的网格任务调度模型及算法研究 [J]. 计算机科学, 2013, 40(3): 136-141
- [6] Li Xin, Jia Zhi-ping, Ju Lei, et al. Energy Efficient Scheduling and Optimization for Parallel Tasks on Homogeneous Clusters [J]. Chinese Journal of Computers, 2012, 35(3): 591-602 (in Chinese)
李新, 贾智平, 鞠雷, 等. 一种面向同构集群系统的并行任务节能调度优化方法 [J]. 计算机学报, 2012, 35(3): 591-602
- [7] Cotronis Y, Konstantinidis E, Louka M A. A comparison of CPU and GPU implementations for solving the Convection Diffusion equation using the local Modified SOR method [J]. Parallel Computing, 2014, 40(7): 173-185
- [8] Konstantinidis E, Cotronis Y. Graphics processing unit acceleration of the red/black SOR method[J]. Concurrency and Computation, 2013, 25(8): 1107-1120
- [9] Epicoco I, Mocavero S, Aloisio G. The performance model for a parallel SOR algorithm using the red-black scheme [J]. International Journal of High Performance Systems Architecture, 2012, 4(2): 101-109
- [10] Satoa Y, Hino T, Ohashi K. Parallelization of an unstructured Navier-Stokes solver using a multi-color ordering method for OpenMP [J]. Computers & Fluids, 2013, 88(12): 496-509
- [11] Iwashita T, Nakashima H, Takahashi Y. Algebraic Block Multi-Color Ordering Method for Parallel Multi-Threaded Sparse Triangular Solver in ICCG Method [C]//Proceeding of IEEE 26th International Parallel and Distributed Processing Symposium. Shanghai, China, 2012: 474-483
- [12] Wang Chuan-long, Meng Guo-yan, Yong Xue-rong. Modified parallel multisplitting iterative methods for non-Hermitian positive definite systems [J]. Advances in Computational Mathematics, 2013, 38(4): 859-872
- [13] Meng Guo-yan, Wang Chuan-long, Yan Xi-hong. Self-adaptive Non-stationary Parallel Multisplitting Two-Stage Iterative Methods for Linear Systems [J]. Data and Knowledge Engineering, 2012, 7696: 38-47
- [14] Zhang Cheng-yi, Luo Shuang-hua, Xu Zong-ben. On parallel multisplitting block iterative methods for linear systems arising in the numerical solution of Euler equations [J]. Journal of Computational and Applied Mathematics, 2015, 279: 249-260
- [15] Zhang Li-tao, Li Jian-lei, Gu Tong-xiang, et al. Convergence of Relaxed Matrix Parallel Multisplitting Chaotic Methods for Matrices [J]. Journal of Applied Mathematics, 2014, 2014: 1-9
- [16] Zhang Li-li. Two-Stage Multisplitting Iteration Methods Using Modulus-Based Matrix Splitting as Inner Iteration for Linear Complementarity Problems [J]. Journal of Optimization Theory and Applications, 2014, 160(1): 189-203
- [17] Zheng Ning, Yin Jun-feng. Accelerated modulus-based matrix splitting iteration methods for linear complementarity problem [J]. Numerical Algorithms, 2013, 64(2): 245-262
- [18] Bai Zhong-zhi, Zhang Li-li. Modulus-based synchronous two-stage multisplitting iteration methods for linear complementarity problems [J]. Numerical Algorithms, 2013, 62(1): 59-77
- [19] Butrylo B, Tudruj M, Masko L. Parallel SSOR preconditioning implemented on dynamic SMP clusters with communication on the fly [J]. Future Generation Computer Systems, 2010, 26(3): 491-497
- [20] Xu Qiu-yan. A New Parallel Successive Overrelaxation Iterative Algorithm for Poisson Equation [C]//Proceeding of 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Beijing, China, 2011: 295-300