

MPI 环境下基于边界校正方法的海洋模型的并行化

符晓单, 李洪平

(中国海洋大学 信息科学与工程学院, 山东 青岛 266100)

摘要: 基于区域分解技术, 设计了一种针对海洋模型的边界校正方法, 该方法不需要对原有海洋模型进行较大改动, 只需要利用消息传递接口(message passing interface, MPI)编写子区域间的数据传输模块, 即可实现原有海洋模型的并行化。相对于重新开发并行系统, 该方法能够较快把原有模型的串行解决方案转化为并行解决方案, 并且并行化后的海洋模型能够获得较高的加速比。

关键词: 边界校正; 并行化; 区域分解; 消息传递接口(message passing interface, MPI)

中图分类号: TP399 文献标识码: A 文章编号: 1000-3096(2015)05-0062-06

doi: 10.11759/hyxx20140515001

海洋大气模型中涉及很多由非线性偏微分方程组成的海洋大气耦合方程, 很难求出其中绝大部分方程的解析解, 因此, 主要通过计算求解偏微分方程的数值解进行模拟。传统上, 主要通过串行算法进行数值模拟, 如著名的 POM(Princeton Ocean Model), CAM(community atmosphere model)和 FVCOM(finite volume coastal ocean model)。模型时空分辨率的提高对这些控制方程的计算量和计算速度提出了更高的要求, 原有的串行算法很难满足计算量和计算速度的要求。现代并行技术的发展为提高计算效率带来了机遇, 很多海洋大气模型都转向并行版本的开发, 如并行版本 POM^[1-2]、CAM^[3-4]和 FVCOM^[5]。

到目前为止, 开发并行系统可以使用的工具和函数库有 HPF(high performance fortran), OpenMP(open multi-processing), MPI(message passing interface), SMS(scalable modeling system), GPU(graphic processing unit)等。HPF 是一种基于指令的并行开发工具, 如, 大气输送模型 FRAME(fine resolution AMmonia exchange)^[6]的并行化就是基于 HPF 实现的。MPI 是海洋模式并行化过程中最常用的开发函数库之一, 如 WRF(weather research and forecasting model)模式^[7]的并行化, 但是基于 MPI 的模式程序调试过程较复杂, 而且效率低下。混合 MPI 和 OpenMP 编程模式被认为是在 SMP(symmetrical multi-processing)系统中最有潜力的开发范式之一, Osthoff 等^[8]通过混合 MPI 和 OpenMP 实现 OLAM(Ocean-land-atmosphere model)的并行化, 并行化后的模式在 I/O(Input/Output)操作方面体现出性能的提升, 然而混合编程范式的实现过程比较复杂。SMS^[9]

被用来实现 HYCOM(HYbrid coordinate ocean model)的并行化, 较之于基于 MPI 和 OpenMP 的并行化, SMS 体现出性能的提升, 然而, SMS 只适用于规则型的区域分解。GPU 也可以用于海洋模型的并行化, 如 ROMS(regional ocean modeling system)^[10]的并行化, 然而, GPU 的主要应用于图像处理领域。

本文将介绍一种有效并简易实现的并行计算技术(校正方法), 该方法不需要对原有的串行系统进行较大改动, 利用 MPI 基本函数编写数据传输模块即可实现原有模型的并行, 较之于重新开发并行系统, 该方法可以简单实现, 并且可以获得比较高的加速比。

1 并行技术

1.1 海洋模型基本方程

本文所涉及的海洋模型为 GRGO(global reduced gravity ocean)模型^[11], 该模型的海洋部分是扩展了的 1.5 层约化重力模式, 最早由 Cane 在 1979 年开发, 主要用来研究 ENSO。该模式假定海洋被温跃层分为 2 层, 流动只发生在上层(质量、动量、能量守恒), 下层流体静止且无限深。假设上层海洋的密度随温度线性变化, 那么上层海洋的动量方程、连续方程、热力方程可以表达为

收稿日期: 2014-05-15; 修回日期: 2014-09-09

基金项目: 国家自然科学基金(NO. 41275013); 863 计划(NO. 2013AA09A506-4)

作者简介: 符晓单(1985-), 男, 河北衡水人, 博士研究生, 主要研究方向为海洋模型的并行化, Email: fuxiaodan8087@163.com; 李洪平, 通信作者, 教授, Email: lhp@ouc.edu.cn

$$\frac{\partial U}{\partial t} + U \cdot \nabla U + fK \cdot U - \frac{1}{2}h\nabla b + \nabla(hb) = \frac{\tau}{h} + \mu \nabla^2 U - \frac{1}{h}w_e H(w_e)U \quad (1)$$

$$\frac{\partial h}{\partial t} + \nabla \cdot (hU) = w_e \quad (2)$$

$$\frac{\partial \tau}{\partial t} + U \cdot \nabla T = \frac{Q}{\rho_0 C_p h} + \kappa \nabla^2 T - \frac{1}{h}w_e H(w_e)(T - T_r) \quad (3)$$

其中, $U=U(x, y, t)$, $h=h(x, y, t)$, $T=T(x, y, t)$ 分别示水平流速, 上层的厚度, 温度, x, y 为水平坐标, t 为时间, w_e 为混合层的卷夹速度, b 为上层浮力, Q 为热流, μ 为水平粘性系数, κ 为扩散系数, τ 为风应力向量, T_r 为参考温度, H 表示阶跃函数, 参数的具体物理意义参见文献[11]流速 U 可以分解为两个相互垂直的 u 分量和 v 分量之和: $U=ui+vj$. GRGO 模型主要根据边界条件通过时间积分求解 U, h, T , 因此在求解方程中主要求解 $\frac{\partial}{\partial t}$, 本文并行算法的主要思想是水平方

向上的区域分解, 在以上方程中主要关注 $\frac{\partial}{\partial x}$ 以及 $\frac{\partial}{\partial y}$ 所涉及的网格点关系。

1.2 边界校正方法

校正方法的基本思路可以分为两步。第一步, 每个子区域向周围扩展形成子区域虚拟边界, 这种子区域虚拟边界可以用来计算子区域内部的网格点, 此步计算出来子区域内部网格点是正确的。第二步是对前一步子区域边界进行校正, 以便进行下一次迭代计算。这种子区域边界进行校正的方法可以通过一维的流速 U 分量方程 u 进行理解。

$$u_i^{n+1} = F(u_{i-1}^n, u_i^n, u_{i+1}^n) \quad i=1, 2, \dots, m; u_0^{n+1}, u_{m+1}^n = g(t) \quad (4)$$

上式中 u 表示流速, i 表示空间点, n 表示时刻, $g(t)$ 表示计算流速 u 的边界条件。现在把计算区域分成两个子区域: $u_{1,i}(i=1, 2, \dots, p)$, $u_{2,i}(i=p+1, \dots, m)$, 现在利用“虚拟边界”条件, 对于第一个子区域假定 $u_{1,p+1}^n = u_{2,p+1}^n$, 第二个子区域假定 $u_{2,p}^n = u_{1,p}^n$, 然后两个子区域利用各自的边界条件进行区域内网格点的计算, 即

$$\begin{aligned} u_{1,i}^{n+1} &= F(u_{1,i-1}^n, u_{1,i}^n, u_{1,i+1}^n), i=1, 2, \dots, p. \quad u_{1,p+1}^n = u_{2,p+1}^n \\ u_{2,i}^{n+1} &= F(u_{2,i-1}^n, u_{2,i}^n, u_{2,i+1}^n), i=p+1, \dots, m. \quad u_{2,p}^n = u_{1,p}^n \end{aligned} \quad (5)$$

在经过第一步的计算之后, 第一个子区域的右边界 $u_{1,p+1}^n$ 和第二个子区域的左边界 $u_{2,p}^n$ 需要进行校正,

如果边界不进行校正, 则在计算 $n+2$ 时刻子区域内部的网格点时会产生错误。因此第二步是进行边界校正, 校正过程只需利用 MPI 函数“MPI_SEND”和“MPI_RECV”进行子区域边界的传递, 即:

$$u_{1,p+1}^{n+1} = u_{2,p+1}^{n+1}, \quad u_{2,p}^{n+1} = u_{1,p}^{n+1} \quad (6)$$

很明显, 这种方法经过校正之后模型的串行与并行计算完全相同, 此方法的优点是实现简便, 模型计算方程的部分只需进行“虚拟边界”条件的设定, 模型消息传递部分只是子区域边界的传递。

1.3 方程的离散化

在海洋模型问题中, 实际使用的两个网格点间距离 Δx 与方程所描述的洋流运动的特征尺度相比是一个非常小的量, 因此微分方程通常用离散差分的方式进行求解。此模型中的微分方程采用显格式求解满足 CFL(Courant Friedrichs Lewy) 条件, $dt/dx < 0.1$ 即可保证数值解的准确性。插分格式空间上采用 C 网格^[12], C 网格是一种交错网格, 如果网格点上定义流函数(压力、温度等), 那么在半网格上定义速度。以公式(1)中的 u 分量动量方程和公式(2)的连续方程为例, 方程差分形式如下:

$$\begin{aligned} &\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + u_{i,j}^n \frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} + v_{i-1/2,j}^n \frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y} - \\ &f_{i-1/2,j}^n - \frac{1}{2}h_{i,j-1/2}^n \frac{b_{i,j}^n - b_{i-1,j}^n}{\Delta x} + \frac{b_{i,j}^n h_{i,j-1/2}^n - b_{i-1,j}^n h_{i-1,j-1/2}^n}{\Delta x} = \\ &\frac{\tau_x}{h_{i,j-1/2}^n} + u \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} \right) - \\ &\frac{1}{h_{i,j-1/2}^n} w_{ei,j}^n H(w_e) u_{i,j}^n \end{aligned} \quad (7)$$

$$\begin{aligned} &\frac{h_{i,j-1/2}^{n+1} - h_{i,j-1/2}^n}{\Delta t} + \frac{u_{i,j}^n h_{i,j-1/2}^n - u_{i-1,j}^n h_{i-1,j-1/2}^n}{\Delta x} + \\ &\frac{v_{i-1/2,j}^n h_{i,j-1/2}^n - v_{i-1/2,j-1}^n h_{i,j-3/2}^n}{\Delta y} w_e \end{aligned} \quad (8)$$

在上述方程中 n 表示时间, i, j 表示位置, 即网格点, 根据微分方程中的 $\frac{\partial}{\partial x}$ 和 $\frac{\partial}{\partial y}$ 的差分形式确定网格点之间的关系, u 分量和连续方程网格点关系如图 1。网格点关系的不同, 可以确定不同的区域分解方案,

图 1(a)中计算 $n+1$ 时刻网格点为 i, j 的 u (黑色实心圆) 需要用到 n 时刻 u 以及 u 周围四个网格点。图 1(b)中计算 $n+1$ 时刻的 h , 则需要 n 时刻的 $h_{i-1, j}, h_{i, j-1}$ 。

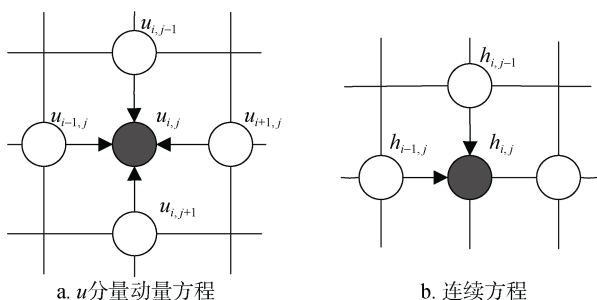


图 1 网格点之间关系
Fig.1 Relationship between grids
箭头代表依赖关系
The arrows represent dependencies

1.4 区域分解

确定网格点之间的关系后, 根据校正方法进行区域分解, 并设计子区域间的数据传输方案, 本方法中的子区域数据传输方案如图 2 所示, 把原有区域分解为若干个子区域之后, 每个子区域向周围延伸一个网格点形成子区域边界, 子区域的内部网格点计算得到的值是正确的。在第二个阶段子区域的延伸边界的网格点通过数据传输函数从周围子区域取得正确的计算结果, 以便进行下一次区域内部网格点的计算。

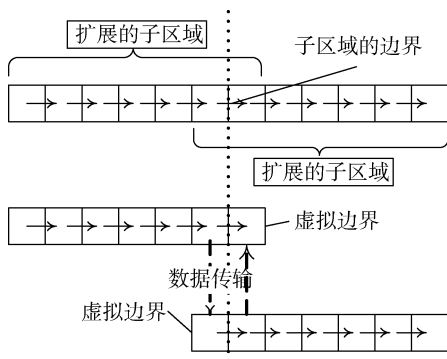


图 2 区域分解和校正方法
Fig. 2 Domain decomposition and correction method

比较规则的区域分解如图 3 所示, 把整个区域分解为 4 个子区域, 第一个子区域在计算之后需要校正的网格单元在子区域扩展之前的边界和子区域扩展之后的边界之间(图中灰色实线与虚线之间)。这些需要校正的网格单元通过周围子区域内部网格点进行校正。

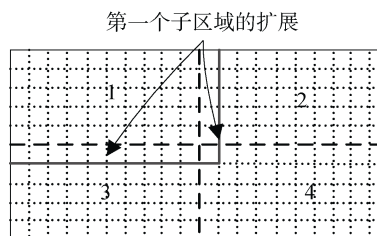


图 3 区域分解
Fig. 3 The domain decomposition

2 并行实现

基于 MPI 环境的并行编程模式是目前海洋模型并行化的主流模式, MPI 编程模式可扩展性好, 适合于集群。本文的核心是采用边界校正方法实现海洋模型的并行化, 由于并行化后的 GRGO 海洋模型需要在集群环境中与大气模式进行耦合, 所以本文在 MPI 环境中实现其并行化。

GRGO 海洋模型的实现流程图如图 4 所示, 首先所有进程为本进程所用到的数组分配内存空间, 并初始化常量以及设置网格点, 进程 $P_i(i=0)$ 负责整个模型的初始化, 主要包括从文件中读取海表面温度和地形等数据, 然后计算并广播发送到其他各个进程。初始化过程完成之后, 所有进程调用 `mpi_barrier` 进行进程同步。下一步是所有进程进入主循环, 系统所有进程读取海洋大气耦合数据, 并根据子区域的边界条件和控制方程的差分形式依次求解 u, h, T , 并行进边界的校正。进程 $P_i(i=1, \dots, n-1)$ 发送本进程计算出来的 u, h, T 到进程 $P_i(i=0)$, 并由此进程进行输出。

3 性能测试及分析

海洋模型并行化的目的是缩短模型的运行时间, 提高模型计算效率, 使得模型满足实验和业务化对模型计算速度的要求。本文首先在以下环境中对边界校正算法进行测试, 硬件环境: 服务器为一台 IBM 3650 M2, CPU 为包含 16 核的 Intel CPU, 并行实现过程中, 1 个 CPU 核对应 1 个并行进程, 软件环境: 操作系统为 RedHat Enterprise Linux 6.0, MPI 的版本为 Open MPI 1.6.5。并行 GRGO 的计算区域纬度范围 $60^{\circ}\text{S} \sim 60^{\circ}\text{N}$, 经度范围 $0^{\circ} \sim 360^{\circ}$, 水平网格大小为 $0.1^{\circ} \times 0.1^{\circ}$, 本文统计海洋模型时间积分迭代 10 次的运行时间。通过在以上环境下测试, 统计出在 CPU 不同核数下模型的运行时间(表 1)。

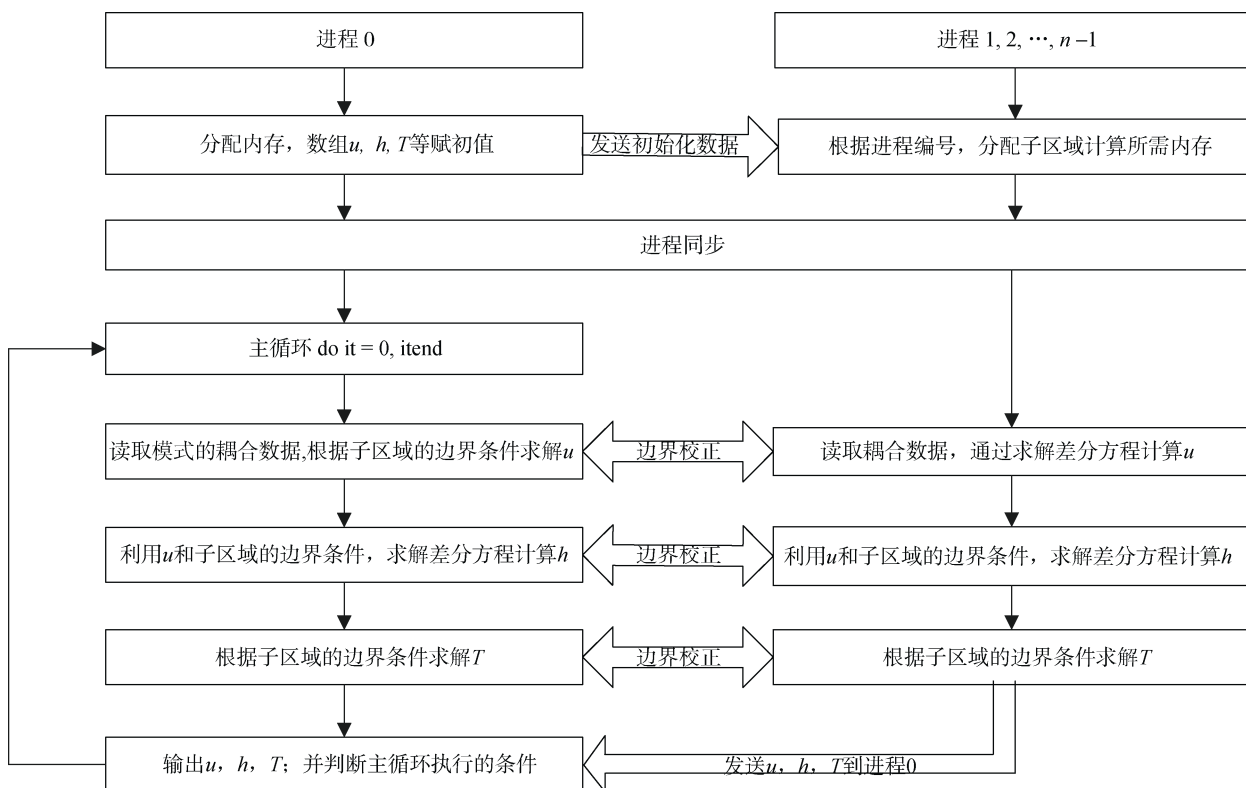


图 4 并行模型流程图

Fig.4 The process of parallelization

表 1 服务器环境下模型的运行时间

Tab.1 Overall timing of the ocean model under server environment

CPU 核数(个)	模型运行时间(s)
1	175.86
2	95.01
4	48.53
6	32.72
8	24.82
10	28.42
12	24.19

根据 CPU 核数和模型的运算时间计算并行化后的加速比, 加速比^[13]=串程序的执行时间/并程序的执行时间, 结果如图 5 所示。由图 5 和表 1 可以看出, 随着 CPU 核数的增加, 采用边界校正算法并行化后模型的运行时间和计算时间逐渐减小, 模型的加速比逐渐增加, 在 CPU 核数为 12 时, 模型的运行时间和计算时间降低到最低分别为 24.19 s 和 20.10 s, 并且并行化后的加速比达到最大值 7.27, 这表明, 增加 CPU 核数可以获得较高加速比。

为了测试边界校正方法的可移植性, 本文在机

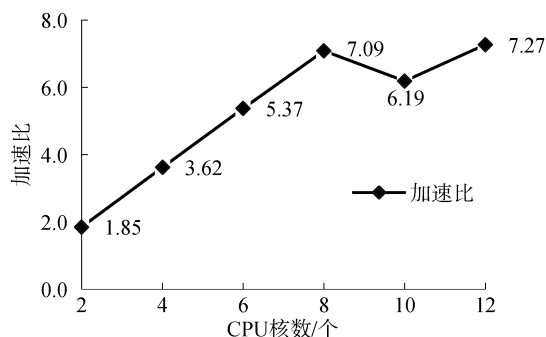


图 5 服务器环境下并行加速比

Fig.5 The speedup of the parallelization under server environment

群环境下对边界校正方法进行测试。本文侧重于并行算法的设计与分析, 由于大规模的海洋模型涉及耦合模块以及大气模式, 分析过程较为复杂, 所以本文只统计模型计算过程中 u 分量方程的计算时间。

根据表 2 中的 CPU 核数以及计算时间, 我们可以计算出机群环境下模型求解 u 方程的加速比(图 6)。由表 2 和图 6 可知, 随着计算节点的增加, 模型计算 u 分量方程的计算时间逐渐减少, 加速比逐渐增加。在 CPU 核数为 120 时, u 分量方程的计算时间

为 0.49s, 此时加速比为 38.2。测试结果表明, 增加 CPU 核数, 可以使计算 u 分量方程的时间减少, 得到较高的加速比, 因而可以满足实验和业务化对模型计算速度的要求。

表 2 机群环境下 u 分量方程时间
Tab. 2 Overall timing of solving u equation under multi-nodes

CPU 核数(个)	u 分量方程计算时间(s)
1	18.71
12	4.58
24	2.68
36	1.67
48	1.13
60	0.90
72	0.77
84	0.64
96	0.51
108	0.51
120	0.49

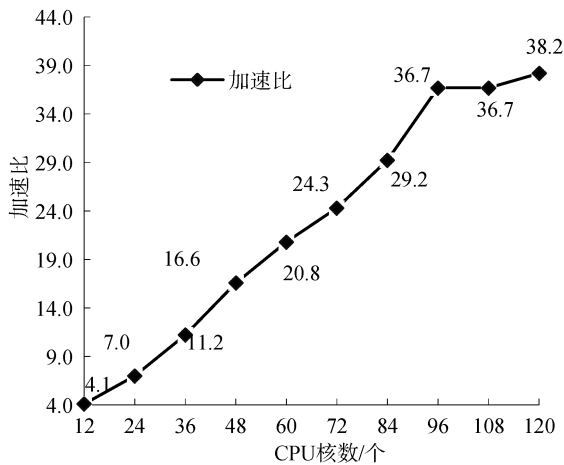


图 6 机群环境下并行加速比

Fig.6 The speedup of the parallelization under multi-nodes

4 结论

本文设计了一种基于区域分解的边界校正方法, 采用该边界校正方法实现了海洋模型 GRGO 的并行化, 在两种并行环境下对并行化后的 GRGO 模型进行分析: 多核服务器环境下, 随着 CPU 核数的增加, 并行化后的海洋模型的运行时间逐渐减少, 获得的加速比逐渐增大; 在 CPU 核数为 12 时, 加速比为 7.27。在机群环境中得到测试结果与多核服务器下的结果较为类似, 在 CPU 核数为 120 时, 得到的加速

比为 38.2。并行化后的海洋模型在保证计算结果正确性的前提下, 能够满足实验和业务化预报的需要。

参考文献:

- [1] Jordi A, Wang D P. sbPOM: A parallel implementation of princeton ocean model[J]. Environmental Modelling & Software, 2012, 38: 59-61.
- [2] Giunta G, Mariani P, Montella R, et al. POM: A nested, scalable, parrallel and Fortran 90 implementation of the princeton Ocean Model[J]. Environmental Modelling and Software 2007, 22(1) : 117-122.
- [3] Mirin A A, Worley P H. Improving the performance scalability of the community atmosphere model[J]. International Journal of High Performance Computing Applications, 2012, 26(1): 17-30.
- [4] Tseng Y H, Ding C. Efficient parallel I/O in community atmosphere model (CAM)[J]. International Journal of High Performance Computing Applications, 2008, 22(2): 206-218.
- [5] Cowles G W. Parallelization of the FVCOM coastal ocean model[J]. International Journal of High Performance Computing Applications, 2008, 22(2): 177-193.
- [6] Pais V A, Fournier N, Sutton M A, et al. Using high performance fortran to parallelise a multi-layer atmospheric transport model[J]. Parallel Computing, 2004, 30(1): 21-33.
- [7] Wu Y, Xu G, Zhao Y, Tan Y. Parallel processing on WRF meteorological data using MPICH[C]//Proceedings-6th International Conference on Internet Computing for Science and Engineering. United States: IEEE Computer Society, 2012: 262-265.
- [8] Osthoff C, Grunmann P, Boito F, et al. Improving performance on atmospheric models through a hybrid OpenMP/MPI implementation[C]//Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications. United States: IEEE Computer Society, 2011: 69-74.
- [9] Govett M, Hart L, Henderson T, et al. The scalable modeling system: directed-based code parallelization for distributed and shared memory computers[J]. Parallel Computing, 2003, 29(8): 995-1020.

- [10] Mak J, Choboter P, Lupo C. Numerical ocean modeling and simulation with CUDA[C] // MTS/IEEE Kona Conference, OCEANS'11. United States: IEEE Computer Society, 2011: 19-22.
- [11] Fang Y. A coupled model study of the remote influence of ENSO on tropical atlantics SST variability[D]. Texas A&M University, 2005.
- [12] 夏长水, 陈显尧, 乔方利, 等. C 网格嵌套技术及其在海洋波动传播模拟中的应用[J]. 海洋科学进展, 2003, 21(4): 401-406 .
- [13] Quinn M J . MPI 与 OpenMP 并行程序设计(C 语言版)[M] . 北京: 清华大学出版社, 2004: 134-136 .

A boundary correction method for the parallelization of ocean model under MPI environment

FU Xiao-dan, LI Hong-ping

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

Received: May, 15, 2014

Key words: boundary correction; parallelization; domain decomposition; MPI(message passing interface)

Abstract: Based on domain decomposition, this paper proposes a parallel technology for ocean model, and this technology doesn't need many changes of the original ocean model, which can parallelize the original ocean model only by using MPI basic functions to construct data transmission module between sub-domains. Relative to redevelopment of a new parallel system, the parallel technology can quickly transform the serial solution of the original model into parallel solution, and the parallel version of the ocean model can get a higher speedup.

(本文编辑: 李晓燕)