

一个数值天气预报模式的并行混合编程模型 及其应用*

朱政慧

(国家气象中心数值室 北京 100081)

摘 要

提出了一个适用于数值天气预报模式的并行混合编程模型, 归纳出混合并行算法设计的特点. 依据该模型, 针对国内发展的区域数值天气预报模式, 开发基于 MPI/OpenMP 的并行版本. 通过对五种并行方案的并行性能分析比较, 发现其结果同抽象模型的分析结论是一致的.

关键词: 混合编程, 数值天气预报, 并行性能

A PARALLEL HYBRID PROGRAMMING MODEL FOR NUMERICAL WEATHER PREDICTION MODELS AND ITS APPLICATION

Zhu Zhenghui

(Numerical Weather Prediction Department, National Meteorological Center, 100081)

Abstract

In this paper an analytical mathematical model suitable for numerical weather prediction models is developed. Actual computations are performed on HLAFS, a mesoscale grid point NWP model developed in China. On the IBM RS6000 SP. The forecast model is parallelized using the hybrid MPI/OpenMP programming method, in combination with various domain partition techniques. The obtained results are analyzed and found to be in good agreement with the analytical performance model.

Key words: Hybrid programming, numerical weather prediction, parallel performance

§1. 引 言

数值天气预报模式的并行性能不仅依赖于模式的动力框架和物理过程的算法, 也取决于所运行平台的体系结构. 当前并行计算机系统都是将多个计算结点由一个内部网络连接, 而每个结点内又有多个处理器共享内存, 称之为多结点多处理器的层次结构. 在这种系统上,

* 2003 年 3 月 10 日收到.

采用 MPI/OpenMP 混合编程的并行算法也日趋流行, 已有几个数值天气预报模式如 IFS^[1] 和 WRF^[2,3] 系统都采用了这种并行方法. 然而, 由于气象数值模式这一特殊应用模型自身的特点给并行计算带来的诸多问题, 使人们还不是很清楚各因素独立作用和综合作用的特征及后果. 并行应用模型及其实现的平台各不相同, 采用的混合编程的实施方案也不同, 使得最终的并行性能分析结论也很不一致. 在关于气象数值预报模式的并行计算设计的研究中, 更有科学和实用意义的是根据气象数值模式的基本特点, 对其并行方案设计进行一般性的研究, 从而总结出有普遍指导意义的最优原则和判据, 使气象工作者们在进行并行模式的设计时知道如何进行数据划分、如何平衡各种开销, 获得较好的并行运算性能.

本文提出了一个抽象的并行混合编程模型, 可以用来辅助设计和分析气象模式及有相同计算特性的工程应用程序的并行性能, 预测它们随不同的数据分区映射方案和所参与计算的处理器数间的变化关系. 根据该模型的初步结论, 针对国内自行开发的一个中尺度格点模式 HLAFS, 设计了五种 MPI/OpenMP 编程方案, 实现了其并行计算. 在 IBM SP 并行机上进行了并行计算实验, 发现其结果同抽象模型的分析结论是一致的.

文中第二节介绍了这个抽象模型, 第三节中介绍试验对象 HLAFS 模式和试验平台 IBM SP 并行计算机系统. 对 HLAFS 模式所采用的五种并行计算方案在第四节中予以说明. 在第五节中给出并行计算试验结果及其分析, 第六节为结论.

§2. 一个适用于数值天气预报模式的并行混合编程模型

2.1 数值天气预报模式的计算特性

所谓数值天气预报, 就是在给定初始条件和边界条件的情况下, 数值求解大气运动基本方程组, 由已知的初始时刻的大气状态预报未来时刻的大气状态. 由于该方程组是一组非线性方程组, 通常不可能求得它的解析解, 需要借助计算机采用数值方法求其近似解^[4].

数值求解大气运动基本方程组有两种方法, 一种是差分方法 (即格点法), 另一种是谱方法, 分别得到两种积分模式: 格点模式和谱模式. 前者将空间连续的场用一组离散格点上的值表示出来, 后者则用一组有限截断的正交基函数来展开空间连续的场, 从而将空间连续的状态变量转换为有限个展开函数. 在谱模式的计算中, 包括格点空间的计算、富利叶变换、勒让得变换和谱空间的计算, 其中所有的物理过程和时间滤波等计算都是在格点空间完成的, 占整个模式计算的大部分时间. 格点模式的并行运算是谱模式并行方案设计的重要基础, 从当前的发展来看格点模式也比谱模式具有优势, 所以本文将局限在格点模式的并行算法研究上.

连续的场经空间离散后, 每个三维格点都有一个支持预报值计算的数据域, 即为了计算该格点上的下一时刻预报, 需要用到围绕其周围一定范围内上一时刻或更前时刻的数据, 此数据域的大小依赖于具体的空间差分格式. 它在水平方向很小, 一般情况下不会超过 2-3 个格距. 在垂直方向上, 由于非绝热过程的计算可能涉及垂直积分, 这个数据域可能扩展到整个气柱.

数值预报计算的以上数据域特征, 使得水平区域分解成为数值预报模式并行计算方案的一个基本选择. 另外水平区域分解, 可以保证绝热模式的负载平衡. 在非绝热条件下, 由于

物理过程的差异主要在水平方向, 因此要实现负载平衡, 需要进行水平区域的动态调整.

在多结点多处理器的并行计算机平台上, 采用消息传递并行方式需要在程序中显式说明数据分解映射实施方案, 共享变量并行, 数据划分是由系统隐式完成的. 不论采用何种通信方式, 通信开销、同步开销和负载不平衡开销都同数据结构的划分方案密切相关的. 通过建立相应的数学公式模型, 我们可以评估数值模式的并行执行时间, 为并行方案设计选择最好的分解方法.

2.2 并行混合编程模型

根据黄铠和徐志伟提出的阶段并行模型^[5], 在一个超步(单位时间步)内, n 个处理器并行求解一个应用问题所需花费的时间由三部分组成: 计算阶段执行时间 T_{comp} , 交互时间 $T_{interact}$ 和并行时间 T_{par} . 由以下公式表示:

$$T_{comp} = (w + \sigma\sqrt{2\log n})t_f, \quad (1)$$

$$T_{interact} = t_0(n) + \frac{m}{r_\infty(n)} = t_0(n) + m \cdot t_c(n), \quad (2)$$

$$T_{par} = t_p(n), \quad (3)$$

其中 w 和 σ 分别代表每个处理器工作负载的平均值和标准偏差, t_f 代表执行计算操作的平均时间, m 表示消息长度. $t_0(n)$ 为启动时间(延迟), $r_\infty(n)$ 为渐进带宽, $t_c(n)$ 为每字节消息传送时间, $t_p(n)$ 为并行开销, 都是机器规模 n 的函数.

在数值天气预报模式中, 将大气模型看作是一个长方体, 经离散后东西方向的格点数是 $I0$. 南北方向上的格点数是 $J0$, 垂直层次是 $K0$, 积分时间步长为 dt , 经 T/dt 步计算得到 T 时刻的预报值. 采用均匀区域分解的方法, 对水平数据结构进行二维数据划分. 设东西向由 NX 个处理器并行处理, 南北向由 NY 个处理器并行处理, 则参与计算的总处理器数为 $n = NX \times NY$. 定义参与 I 方向并行计算的处理器数 $NX = NX_{mpi} \times NX_{omp}$, 而参与 J 方向并行计算的处理器数 $NY = NY_{mpi} \times NY_{omp}$, 其中 NX_{mpi} , NX_{omp} , NY_{mpi} 和 NY_{omp} 分别代表在 I 方向和 J 方向参与计算的多进程和多线程的配置数.

若将每个积分时间步作为一个超步, 则可对 (1), (2) 和 (3) 式作进一步推导.

在计算阶段, 每个处理器的平均计算量为每个分区内的格点总数, 即 $w = K0 \times \left(\frac{I0}{NX} \times \frac{J0}{NY} \right)$. 这时 t_f 表示每个处理器用于计算单位格点所花的平均时间. 该阶段的总执行时间应是 n 个处理器各自执行时间中的最大值, 那么 σ 的大小应为 n 个处理器中完成计算量最多的那个处理器的计算量比平均计算格点数多出的部分. 每个周期内计算阶段的时间为

$$T_{comp} = K0 \times \left(\frac{I0}{NX} \times \frac{J0}{NY} + \sigma\sqrt{2\log n} \right) t_f. \quad (4)$$

交互阶段, 根据所采用的通信方式不同, 认为消息传递过程的同步开销隐式完成, 不必另外单独计算, 由于分南北和东西双向, 所以启动消息传递的次数不同, 记为 M_{mpi} , 因此总的延迟时间为 $M_{mpi} \times L_a$. 在每层上需要进行消息传递的变量长度实际上为各 MPI 分区的边长之

和, 即 $A \times \frac{I0}{NX_{mpi}} + B \times \frac{J0}{NY_{mpi}}$, A 和 B 的值为 0, 1 或 2, 代表在 MPI 分区内需要进行消息传递的次数. 由于每种变量依类型不同, 所占用的内存字节长度也不同 (如一般单精度实型量是 4 个字节, 双精度实型量是 8 个字节), 所以记 C_{byte} 为每个变量所占用的字节长度, 则在整个计算域上需进行消息传递的总字节长度为 $m = \left(A \times \frac{I0}{NX_{mpi}} + B \times \frac{J0}{NY_{mpi}} \right) \times C_{byte} \times K0$. 共享变量方式的同步开销在每个并行区的结束处产生, 不可忽略, 记为 $t_{syn} \times M_{omp}$. 其中 M_{omp} 为多线程并行区的启动次数, t_{syn} 表示同步时间. 这样交互阶段的时间计算公式为

$$T_{interact} = M_{mpi} \times L_a + \left(A \times \frac{I0}{NX_{mpi}} + B \times \frac{J0}{NY_{mpi}} \right) \times C_{byte} \times K0 \times t_c(n) + t_{syn} \times M_{omp}. \quad (5)$$

并行时间指的是并行化开销, 它应该包括进程创建和分组、线程生成和结束等方面的系统开销时间. 实际上在模式运行过程进行 MPI 进程创建和分组 (即初始化和结束) 的操作只有一次, 记时间开销为 t_{mpi} . 对每个积分时间步, 则有

$$T_{par} = t_{omp} \times M_{omp}, \quad (6)$$

其中 t_{mpi} 表示系统进行 MPI 进程创建和分组 (即初始化和结束) 的开销时间, 而 t_{omp} 表示线程生成和结束的开销时间, M_{omp} 则代表应用程序中要求生成 / 结束多线程组的次数.

根据 (4)-(6) 式, 假定数值模式对每个格点的积分计算量都是均衡的如框架模式, 输入 / 输出时间忽略不计, 计入 t_{mpi} , 则由 n 个处理器并行计算作 T 时刻的物理量场预报, 总的运转墙钟时间应为

$$\begin{aligned} T_n &= T_{comp} + T_{interact} + T_{par} \\ &= \left[K0 \times \left(\underbrace{\frac{I0}{NX} \times \frac{J0}{NY}}_a + \underbrace{\sigma \sqrt{2 \log n}}_b \right) t_f + \underbrace{M_{mpi} \times L_a}_c \right. \\ &\quad + \underbrace{\left(A \times \frac{I0}{NX_{mpi}} + B \times \frac{J0}{NY_{mpi}} \right)}_d \times C_{byte} \times K0 \times t_c(n) \\ &\quad \left. + \underbrace{t_{syn} \times M_{omp} + t_{omp} \times M_{omp}}_e \right] \times \frac{T}{dt} + t_{mpi}. \end{aligned} \quad (7)$$

公式 (7) 就是本文提出的适用于数值天气预报模式的并行混合编程模型. 在忽略气象数值模式的输入 / 输出时间的基础上, 该模型考虑了包括多进程管理和多线程管理在内的并行开销 (e 后半部分)、通信开销 (c 和 d)、同步开销 (e 前半部分) 和负载不平衡开销 (b). 可用于评估和分析实际的并行气象数值模式算法和时间开销. 当 $e = 0$ 时, 该模型就简化为一个纯消息传递模型.

若负载是均衡的, 即 $\sigma = 0$, 当多线程的生成 / 结束及同步开销 (e) 相对消息传递所需的通信时间 (c 和 d) 较小时, 并行模式的最优算法主要由消息传递的分区映射方式决定, 启动消息传递的次数越少、通信字节长度越短, 算法最优. 随着线程数的增多, 需要进行同步

的开销增大,使这一开销的增长速度小于消息传递通信时间随通信量增长的速度,才能保证多进程多线程的混合同并行化编程算法最优.因此,在多结点多处理器的并行计算机平台上设计 MPI/OpenMP 混合同并行数值模式, MPI 任务数和 OpenMP 线程数的划分配置存在一定的相互制约.

§3. 试验对象与平台

3.1 试验对象

在选用实际应用模型——预报模式时,考虑到由于在模式网格设计、时间积分方案、离散方案、垂直坐标、计算技术、分辨率的大小及格点数的选择上都存在很大的差别,难以确定一个综合的最优的试验模型,所以本文在现成的业务模式 HLAFS 的基础上,讨论、比较和选择“最优”算法.优点是实验目标明确,成熟稳定,易于对产生的问题分析定位,缺点是容易受到已编制程序框架的制约.

高分辨率有限区域同化预报系统 HLAFS 是国家气象中心的现有业务模式之一.该系统在物理过程和参数化方案上几经改进,发展成为一个对 $15 \sim 64^\circ \text{N}$, $70 \sim 145^\circ \text{E}$ 范围进行暴雨预报的系统.它包括四部分:分析和初始化部分、预报模式和后处理部分.本文的研究在对预报模式并行化的基础上进行.

HLAFS 预报模式是一个静力平衡中尺度格点模式,采用 Arakawa C 型经纬网格点.在空间和时间离散方案上均使用了有限差分方案.模式采用显式的蛙跃时间积分方案,同时结合时间滤波抑制虚假的计算波的增长.垂直方向采用 σ 坐标,边界条件由全球预报模式提供.物理过程包括通量、热量和水气的水平扩散、积云对流参数化、土壤过程、垂直湍流扩散、辐射和格点可分辨尺度降水过程等 [6].

目前,模式可以在两种水平分辨率下运行:一种是 $0.5^\circ \times 0.5^\circ$,计算格点量为 $I_0 \times J_0 \times K_0 = 181 \times 119 \times 20$,时间步长为 30 秒,其中 I 为经向格点数, J 为纬向格点数, K 为垂直层数.为适应新时期的预报要求,另一种分辨率为 $0.25^\circ \times 0.25^\circ$,计算格点量为 $I_0 \times J_0 \times K_0 = 361 \times 237 \times 20$,时间步长为 15 秒,并引入了新的包括冰相过程的显式降水方案,计算量显著加大.

3.2 试验平台

本文所使用的并行计算机是国家气象中心的 IBM RS6000 SP 系统,它是目前国际上高性能计算机的主流机型之一,包括 2 个 I/O 结点和 10 个计算结点.属于 Night Hawk1(简称 NH1)系列. I/O 结点的 CPU 主频为 332MHz.应用程序的并行计算由计算结点完成.每个计算结点内有 8 个 POWER3 222MHz 处理器芯片,由一个共享内存交叉开关相连,采用了对称多处理器结构 (SMP, Symmetric Multi-Processor).不同的结点间通过高性能开关通信.所以,在每个 SMP 结点内处理器之间的通信可通过共享内存完成,而一个 SMP 结点不能直接访问另外一个结点的内存,必须通过消息传递实现.

操作系统为 AIX 4.3.3,编译器 V7.1 版.在应用程序并行编程接口方面,可用 MPI, PVM, OpenMP 和 HPF 等.结点间的通信通过 MPI 来完成,而结点内既可用 MPI,也可用 OpenMP.因此在程序设计上有很大的灵活性,适用于本文的研究.

§4. 试验实施方案设计

对给定的模式，时空差分算法已定，在确定的两种数据集大小情况下，内存需求也基本固定，I/O 方式保持不变，要想获得好的性能，根据前面发展的抽象模型，研究重点应集中在数据映射方案（并行算法）、通信、粒度、负载平衡等方面。

在 IBM RS6000 SP 并行计算机上，在水平区域分解的基础上，设计并实施了五种 MPI/OpenMP 并行 HLAFS 预报模式计算方案。

方案一如图 1 所示，以参与计算的 CPU 数 $n = 8$ 为例，并且 $NX = NX_{omp} = 2$ ， $NY = NY_{mpi} = 4$ 。首先对模式进行 J 方向的 MPI 分区映射，让不同的 MPI 任务 (task) 共同完成各自分区行内的格点计算，由 0 号 CPU 进行 I/O 处理。由于 $0.5^\circ \times 0.5^\circ$ 模式 $J0=119$ 和 $0.25^\circ \times 0.25^\circ$ 模式 $J0=237$ ，大部分情况下不能在多处理器间平均分配，因此将较多的计算量平均分配给行序号较大的任务执行。这样，0 号任务所分得的计算量一般都少于或等于其它任务的计算量，所以由它来进行输入输出处理，这样减少多任务间的负载不平衡。OpenMP 并行指导语句添加在各循环体最外围，用来实现沿纬圈方向 (I 方向) 的并行处理，使得每个任务可由节点内的多个 CPU 并行执行，称为多线程 (thread)。每个 MPI 任务分区内的数据又按列分配给 OpenMP 多线程并行处理。此时，数据是被隐式分配的，运行时系统根据作业卡中指定的环境变量确定数据分配方式。

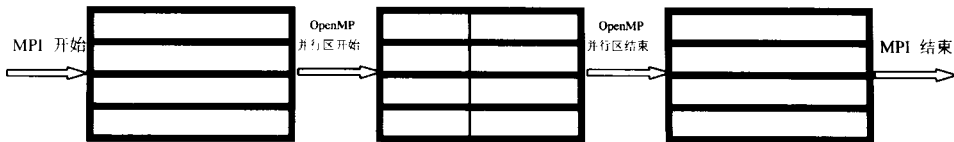


图 1 方案一数据映射流程 $n = 8$, $NX = NX_{omp} = 2$ 且 $NY = NY_{mpi} = 4$

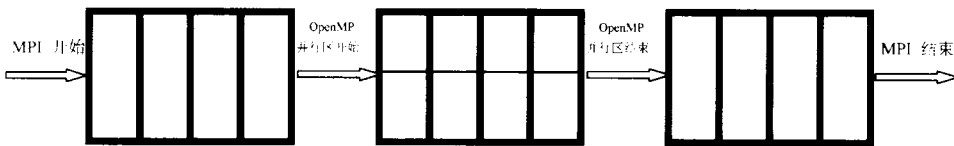


图 2 方案二数据映射流程 $n = 8$, $NX = NX_{mpi} = 4$ 且 $NY = NY_{omp} = 2$

方案二类似于方案一，但是在 I 方向进行 MPI 数据分解，而在 J 方向用 OpenMP。其示例如图 2 所示。这样混合编程采用一维 MPI 数据映射，编程会比较简单。在方案三中，还是采用了传统的二维数据分解的方法。如图 3 所示，通信完全由 MPI 完成。

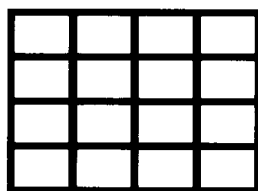


图 3 方案三数据映射流程 $n = 16$, $NX = NX_{mpi} = 4$ 且 $NY = NY_{mpi} = 4$

方案四和方案五中的混合编程同前面方案一和方案二采用的方式有所不同. 它们是在方案三的基础上, 先进行二维分解, 再由 OpenMP 线程来参加 I 方向或 J 方向的计算, 其数据流分别如图 4 和图 5 所示.

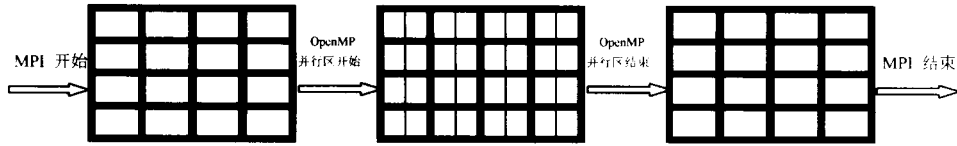


图 4 方案四数据映射流程 $n = 32$, $NX_{mpi} = 4$, $NX_{omp} = 2$ 且 $NY = NY_{mpi} = 4$

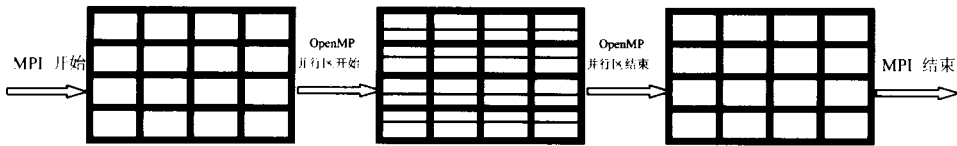


图 5 方案五数据映射流程 $n = 32$, $NX = NX_{mpi} = 4$, 且 $NY_{mpi} = 4$, $NY_{omp} = 2$

§5. 并行计算试验与分析

为对已有的串行应用程序的并行化处理, 首先应对其进行优化, 尽管编译器可以对源程序进行大部分优化工作, 但人工优化仍然是必不可少. 之后的一项重要工作是分析应用程序结构, 明确运算耗时集中在哪些部分, 依此确定合适的并行计算方案及并行程序接口. 然后才是进行任务分解, 实现处理器间的通信, 完成并行计算. 最后, 调试、监测和分析并行程序的执行. 本文试验中, 对 HLAFS 的优化和并行化即遵循了这一原则.

对串行 HLAFS 模式的改进和优化, 主要工作集中在以下四方面: 向量代码到标量代码的转换和高速缓存的优化使用; 算术运算转换; 减少模式的内存需求; 开发用户友好的界面. 对这部分内容的详细介绍, 请参见文献 [6]. 根据本文前面介绍的并行试验方案设计, 完成了模式的并行编程. 通过在 8 个结点的 SP 并行机上运行高、低两种分辨率的 HLAFS 模式, 可以得到上述五种方案的并行效率. 考虑到在不同的气象背景条件下模式的运行特征会有所不同, 尤其是针对不同的物理过程 (如辐射、积云对流、陆面和边界层影响等) 情况更为复杂. 国外学者研究表明 [8], 物理过程部分对并行模式的性能影响很大, 特别是对流和辐射部分, 容易造成负载不平衡. 为了研究并行算法中各因素的独立作用, 分别对关闭物理过程的动力框架部分和打开物理过程后的模式进行了测试. 使用了两种资料, 低分辨率模式所用的个例为 2000 年 7 月 18 日 00 世界时的资料, 而高分辨率模式所用的个例为 1998 年 6 月 26 日 00 世界时的资料. 前者模拟了受台风影响而造成的华南地区的强降雨, 后者模拟了梅雨期间中尺度雨团造成的长江中下游地区的暴雨.

由于参与计算的总处理器数 $n = NX \times NY = NX_{mpi} \times NX_{omp} \times NY_{mpi} \times NY_{omp}$, 有不同的任务数和线程数配置方式, 由此也可以得到不同的墙钟运行时间. 并行计算试验中, 测试了在 $n = 2, 4, 8, 16, 32, 64$ 个处理器参与计算时, 不同配置下框架模式 (关闭物理过程) 和预报模式 (打开物理过程) 分别作 6 小时预报所需运行的墙钟时间. 这里选出各方案

中所对应最佳并行效率进行比较，得到以下对比图：

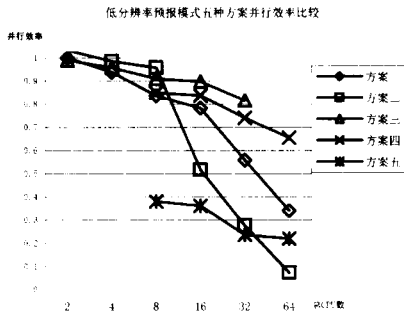


图 6 低分辨率预报模式五种方案并行效率比较图

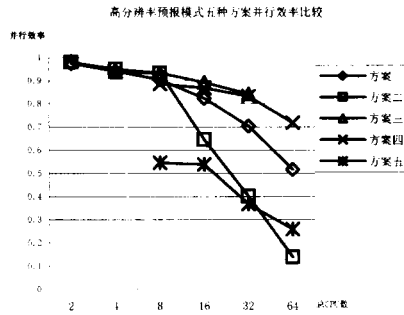


图 7 高分辨率预报模式五种方案并行效率比较图

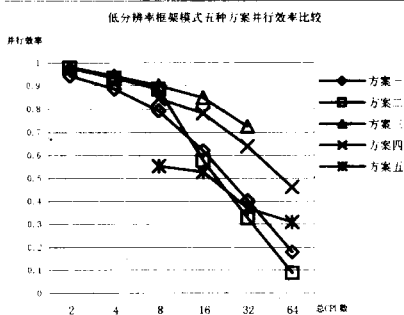


图 8 低分辨率框架模式五种方案并行效率比较图

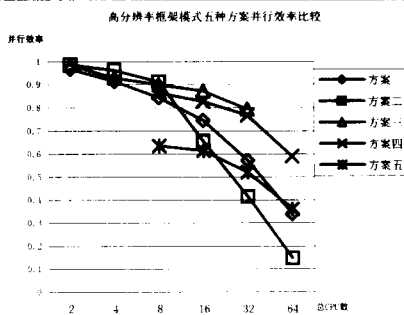


图 9 高分辨率框架模式五种方案并行效率比较图

可以看出，对于 HLAFS 模式，在 IBM SP 并行计算机上，预报模式和框架模式的并行性能基本相似。方案三和方案四最优，方案一居中，方案二和方案五最差。进行模式的水平双向 MPI 划分能获得最好的并行计算性能。尽管本试验所得结果显示混合编程方式（方案四）仍然没有纯 MPI 方式（方案三）高效，但从图上可看出方案四已达到或接近方案三的水平。由于模式边界条件的限制，使得纯 MPI 双向划分在此受到很大限制（最多只能 32 个处理器参与计算），但通过使用 OpenMP，可使更多的处理器参与计算。因此，混合编程是有效的，即在双向 MPI 划分的基础上，结合 OpenMP 指导语句的混合编程能使高分辨预报模式更灵活、更有效的利用多结点多处理器体系结构进行并行计算、向更大规模的并行计算扩展。另一方面，从抽象模型上分析，如果系统提供的 OpenMP 性能（多线程管理及同步开销的控制等）较好的话，混合编程方式应该可以提供最优的并行计算性能。

采用纯粹的结点间用消息传递、结点内用共享内存的方式（方案一）并行性能一般，其处

于劣势的主要原因在于仅对 J 方向进行 MPI 划分使需要进行消息传递的字节长度较大 (d 部分的值较大)。另外, 结点内 8 个线程参与计算时, 需要进行同步的开销过大, 每步的计算性能也不稳定。至于方案二和方案五, 由于在循环体的第二层添加 OpenMP 并行指导语句, M_{omp} 值过大, 使得同步开销和并行开销 (e 部分) 激增, 是这两种方案失败的根本原因。

从计算量对并行效率的影响来看, 除方案三外, 随着模式分辨率的提高和物理过程的加入, 其它各方案的并行效率均明显提高。对于方案三, 可以看出, 分辨率的提高对并行效率的改善不明显, 而只有物理过程的加入使得模式的并行效率有明显提高, 造成这一结果的根本原因在于随着框架模式分辨率的提高, 模式要进行消息传递的字节长度也要随之加大, 传递时间加长, 而加入物理过程最终使通信与计算量之比减小, 并行效率也就提高了。因此, 从这方面分析, 随着计算量的加大, OpenMP 的并行性能的改善程度比 MPI 高。

负载均衡方面, 框架模式的负载比较均衡。而对于物理过程部分, 由于 HLAFS 模式的主要预报范围在陆地上, 海上部分较少, 所以对流过程的负载不平衡较少。另外, 由于采用了较简单的地面过程, 负载也比较均衡。

消息传递过程的时间开销是与参与计算的 MPI 任务数和消息传递的字节长度成正比的。由于一般有 $I_0 > J_0$, 因此 I 方向的任务划分比 J 方向划分所需传递的字节长度小, 因此在 I 方向的任务划分具有优势。同样, 共享变量的时间开销也是与参与计算的 OpenMP 线程数和需要共享的字节长度成正比的。最主要的还是多线程间的同步开销, 线程越多, 开销越大。

§6. 结 论

总之, 在 IBM SP 并行机上, HLAFS 模式向大规模扩展时, 方案三和方案四最优, 方案一居中, 方案二和方案五最差。并行 HLAFS 模式的这些运行特点, 同前面介绍的抽象模型的分析结论是一致的。

从计算与通信的关系来看, 科学计算应用程序的并行处理可分为三类: 一类是各处理器可独立完成计算, 不需要相互间的通信; 第二类是各处理器上的大部分计算是彼此独立的, 但在边界上需要进行信息交换; 第三类则需要密集的通信, 计算较少。气象领域的计算是属于第二类的, 由于当前多结点多处理器的大规模并行计算机的普遍采用, 本文的研究不仅为数值天气预报模式的高性能计算和并行方案设计提供了依据, 也可用于有类似计算特性的科学问题的并行处理。

参 考 文 献

- [1] G Mozdzyński. IFS tests using MPI/OpenMP. ECMWF Newsletter, 88(2000) 13~16.
- [2] J Michalakes, J Dudhia, et al. Design of a next-generation regional weather research and forecast model. Towards Teracomputing, New Jersey: World Scientific, 1999, 117~124.

-
- [3] J Michalakes, S Chen, J Dudhia, et al. Development of a next-generation regional weather research and forecast model. *Developments In Teracomputing*, New Jersey: World Scientific, 2001. 269~276.
 - [4] 田永祥, 沈桐立, 葛孝贞, 陆维松, 数值天气预报教程, 气象出版社, 北京, 1995.
 - [5] K Hwang, Z W Xu. *Scalable Parallel Computing (Technology, Architecture, Programming)*. Beijing: *China Machine Press*, 1999.
 - [6] 朱政慧, 闫之辉, Zaphiris Christidis, 并行高分辨率有限区预报系统在 IBM SP 上的建立, *应用气象学报*, 14:1(2003) 119~121.
 - [7] R. W. Ford, P. M. Burton. Load balancing physics routines. *Towards Teracomputing*, New Jersey: World Scientific, 1999, 147~159.