

OpenMP 在 MM5 中尺度模式 中的应用试验*

张 昕 季仲贞 王 斌

(中国科学院大气物理研究所大气科学和地球流体力学数值模拟国家重点实验室, 北京 100029)

摘 要 简要介绍了一种新的并行编程标准——OpenMP, 通过与其他几种并行编程标准的比较, 了解其优越与不足之处。详细介绍了 OpenMP 在 PSU/NCAR 的中尺度模式 MM5 中的运用。分别用 OpenMP 并行方式与 MPI 并行方式在 SGI Origin 2000 上进行了并行效能测试。测试结果表明: OpenMP 是一种简单、高效的并行编程标准, 非常适合于气象科研与业务领域应用。

关键词: OpenMP; 并行计算; 中尺度模式; 并行加速比; 并行效率

1 引言

在科学与工程计算领域, 有许多问题涉及到大规模计算。譬如, 数值天气预报、气候数值模拟、空气动力学、分子生物学等领域都需要具有高速运算能力的高性能计算机的支持。但是, 目前单处理机运算速度满足不了大规模计算的需求。因此, 人们提出了并行处理的概念。所谓并行处理就是用多个处理器(CPU)来同时处理一个问题, 以提高解题速度。特别是在气象中尺度领域, 由于中尺度现象无论在空间上还是在时间上分辨率都很高, 要求业务数值预报模式或数值研究模式在空间和时间上的分辨率尽量高, 模式运行时间尽量短。因此, 对计算机的计算速度和存储容量要求很高, 发展并行计算是一条切实可行的道路。

随着求解问题规模的不断扩大, 要求的 CPU 数也不断增加, 并行机的可扩展性就显得很重要。而要实现并行机的可扩展, 既要求硬件可扩展, 也要求软件可扩展。从 80 年代开始, 通过可扩展的网络互联技术而形成的分布内存多处理机发展起来, 因为它不需要支持缓存一致, 所以在某种意义上, 它可无限扩展。因此, 作为分布内存处理机的并行编程界面——MPI (Message Passing Interface) 首先发展成熟起来。

随着支持缓存一致的可扩展的共享内存多处理机 (Scalable Shared-Memory Processor, 简称 SSMP) 结构的出现, 1997 年 10 月, 一组计算机硬件和软件厂商联合定义发表了共享内存编程应用程序接口的工业标准建议——OpenMP。目前, 它已得到了包括 SGI/Cray、IBM、DEC、HP、Intel 等计算机厂商和 Kuck& Associates 等软件公司的支持, 使得跨平台可移植、可扩展的共享内存式并行软件的设计成为可能。

2000-09-01 收到, 2000-09-20 收到修改稿

* 国家重点基础研究发展规划项目 G1999032801、国家杰出青年科学基金项目 49825109 和国家自然科学基金资助项目 49975020 的共同资助

2 OpenMP 简介

MPI 将消息传递编程方法标准化、规范化。MPI 已成为消息传递编程的一个可移植的被广泛接受的标准。然而, MPI 难于编程, 它要求程序员将数据结构按不同的处理机分开, 并且, 它也不支持增量并行。而共享内存编程却比较简单, 但不易于扩展。

随着计算机技术的发展, 支持缓存一致的多处理机结构趋于成熟。对于这种系统而言, MPI 就显得非常繁琐。同时, 政府和科学研究部门有大量的 Fortran 77 程序需要并行化。因此, 一种可移植的, 可扩展的共享内存编程界面——OpenMP 应运而生。OpenMP 适合将现存的程序并行化, 同时它依然保持了足够的灵活性以支持除了科学计算以外更广泛的应用。表 1 对几种标准并行编程方法作了一个对比。由表 1 可看到, 克服了难于扩展的缺点以后, OpenMP 不仅能对程序作增量并行转化, 而且也能支持数据并行。所以, OpenMP 与其他的并行编程标准相比, 优越性是显而易见的。

OpenMP 是一组编译指导语句、一组库函数和环境变量的集合, 能用来表示共享内存并行, 可支持粗、细粒度的并行。它包括 3 部分: 控制结构、数据环境和同步机制。另外, 该标准还包括了一个可调用运行时库及相应的环境变量。下面来看一个简单的例子, 了解一下 OpenMP 具体是如何工作的。

```

program main
integer i,sum
sum=0
c234567890
c $ omp parallel do private(i)
c $ omp+reduction(+:sum)
do i=1,50
sum=sum+i+(100-i+1)
enddo
c $ omp end parallel do
print *, '1+2+3+...+99+100 = ',sum
stop
end

```

表 1 OpenMP 与其他几种并行编程标准的对比

	MPI	Pthreads	HPF	OpenMP
可扩性	是	有时	是	是
增量并行	否	否	否	是
可移植性	是	是	是	是
FORTRAN 联编	是	否	是	是
高级别并行	否	否	是	是
数据并行	否	否	是	是

OpenMP: A Proposed Industry Standard API for Shared Memory Programming, <http://www.openmp.org>;
Pthreads: 不是为科技/高性能计算领域而设计的, 极少支持 Fortran 语言, 主要提供较多的任务并行; HPF: 为并行计算机开发的 Fortran 派生语言, 主要适用于分布式内存系统。

上面的程序中以 `c$omp` 开头的语句即为并行指导语句, 一般串行编译时, 被识别为注释语句。程序首先以一个进程开始, 当我们第一次遇到 `PARALLEL` 语句时, 则一个或多个进程产生, 同时对应于每个进程的数据环境也被产生。每个进程的数据环境包括一个私有变量 `i` 和一个归约变量 `sum`, 在并行区中对 `i` 和 `sum` 的引用都是私有的。而归约变量意味着在并行区结束时, 将各个进程的 `sum` 变量以操作符+归约到主进程的 `sum` 变量。在程序执行过程中, 编译器负责将任务分配到不同的进程, 编程者甚至不需要知道到底能用多少个处理器。从上面的例子可以看到, 用 OpenMP 实现并行, 简单易行。对程序员而言, 基本上不必对原来的串行程序做额外的改动和考虑。

3 几个有关并行计算的概念和中尺度气象模式 MM5 的并行化

并行加速比:

$$S_p = t_{\text{seq}} / t_p, \quad (1)$$

其中, t_{seq} 为用串行机完成该任务所需的时间, t_p 为用 p 个处理器并行完成该任务所需的时间。

并行效率:

$$\text{并行效率} = S_p / p, \quad (2)$$

Amdahl 定律:

$$S_p = \frac{1}{s + \frac{1-s}{p}} = \frac{p}{sp + 1 - s} \leq \frac{1}{s}, \quad (3)$$

其中, s 为一特定计算任务中不能并行只能串行执行的运算量占总运算量的百分比, 可并行执行的运算量百分比为 $1-s$, 则并行加速比满足上述定律。其含义就是当 p 趋于无穷大时, S_p 接近 $1/s$, 即不管采用多少台处理机, 可望达到的最大加速比的上限为 $1/s$, s 就是所谓的顺序瓶颈因子。

PSU/NCAR 的 MM5 中尺度气象模式^[1~3]。

MM5 是 1978 年 Anthes 和 Warner 推出的有限区域中尺度气象模式的第 5 代。其主要特点是: (1) 既有静力又有非静力的动力框架; (2) 具有云物理和边界层等多种物理过程; (3) 具有多重嵌套能力; (4) 具有四维资料同化的功能。

MM5 的第 2 版本由 200 多个子程序组成, 包括注释共有约 50 000 行。其全部由标准 FORTRAN 写成, 所以能在大多数平台上运行。由于 PSU/NCAR 的中尺度模式最开始是在 Cray 向量机上运行, 为了提高效率, 对模式进行了向量化处理 (如图 1), 整个模式空间被分成若干南北向的垂直剖面, 即在南北方向进行了向量化处理。因此, 在后来的为共享内存并行机写程序时, 也延续了如上的分块方法, 即 J 循环被分配到了不同的处理器上, 每个处理器担负各自南北向垂直剖面上的运算。

在原有 MM5 串行程序中, 共有 117 处加入了 OpenMP 并行指导语句。它们分别用在计算模式边界条件及边界物理量值, 计算总的干湿空气质量, 插值、对数组的初始

化等。可以发现, 这些子程序都涉及到大量网格点的计算, 特别适合于并行计算。另外, MM5 中最主要的预报程序 SOLVE3 也被并行化, SOLVE3 负责计算诊断量 p^* 、 u 、 v 、 Q_v 、 T 和其他水汽量的倾向值。每向前积分一步, 该子程序将被调用一次。而相似的计算发生在模式几乎所有的格点上。所以, 并行处理这个大循环将显著地缩短运行时间。

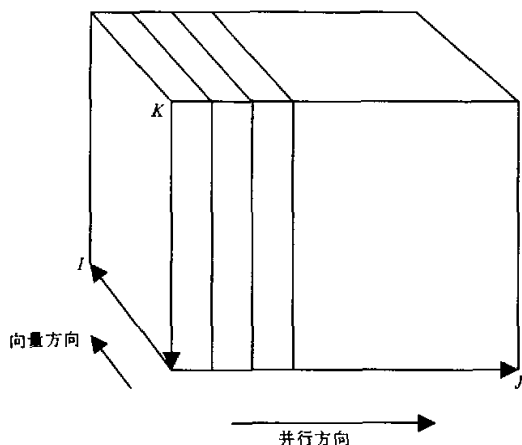


图1 MM5 的数据并行方案

实际上, MM5 所用到的并行指导语句非常简单, 所有这 117 处都是循环级并行, 也即意味着 MM5 的 OpenMP 版本只简单地用到了 PARALLEL DO 语句, 将一些大循环并行化, 并没有用到 OpenMP 中更复杂的概念。在下面的部分, 我们将看到即使如此简单的并行也能收到令人惊喜的效果。

相比之下, MPI 则显得过于繁杂。NCAR 也发布了 MM5 的 MPI 并行版本^[4], 该并行版本由美国 Argonne National Laboratory 的 John Michalakes 完成。他发展了一个称之为 Fortran Loop and Index Converter (FLIC) 的转换器^[5], 该转换器能对一个串行程序作一前处理, 将其转化为 MPI 并行程序。但这并不是一劳永逸的, 其中有关 I/O 处理、嵌套处理等的程序依然需要额外的修改。

4 在 SGI Origin 2000 上的并行计算实验

本文所作的测试均在中国科学院大气物理研究所的 SGI Origin 2000 上进行。该机器采用 20 个 195 MHz 的 IP27 处理器 (MIPS R10000 3.4v) 组成, 带有 2G 内存, 运行 64 位的 IRIX 操作系统, 属于前面提到的可扩展共享内存并行机系统。由于采用了分布式共享内存, 具有 cc-NUMA 结构, 所以可同时支持两种不同的编程方法, 即共享内存编程 (OpenMP) 和消息传递编程 (MPI), 且 O2000 的 MIPSpro 编译器支持自动并行和用户指导并行 (即 OpenMP)。

我们选用 1981 年 7 月 11~12 日四川特大暴雨过程, 分别采用网格距为 60 km, 格点数为 $73 \times 76 \times 23$ 的无嵌套设置和网格距为 60、20 km, 格点数为 $73 \times 76 \times 23$ 、 $76 \times 79 \times 23$ 两重嵌套设置, 分别在 1、2、4、6、8 个 CPU 上运行流体非静力的 MM5。

表 2、表 3 是用 OpenMP 并行的 MM5 模式在 O2000 上运行所得数据。由表 2、表 3 可以看出, 若无嵌套 ($73 \times 76 \times 23$), 随着 CPU 数的增加, 并行加速比呈增长趋势, 用 8 个 CPU 时, 并行加速比达到了 6.88。对于两重嵌套 ($73 \times 76 \times 23$ 、 $76 \times 79 \times 23$), 并行加速比也是随 CPU 数的增加而增加的, 用 8 个 CPU 时, 加速比达到了 7.23。所以, 比较表 2 和表 3 可以发现, 在相同 CPU 数的情况下, 双重嵌套测试所得的并行加速比都大于无嵌套测试所得的加速比。这说明嵌套所增加的运算量大部分都是可以并行的。

根据 (2) 式, 我们计算了不同 CPU 所对应的并行效率 (图 2)。由图 2 可见, 并行效率基本上是随着 CPU 数的增加而降低的, 这主要是因为随着参与计算的 CPU 数的增加, 程序的串行部分越来越成为制约并行效率的瓶颈。这也符合 Amdahl 定律。

为了作一个对比, 我们将无嵌套模式用 MPI 并行方式进行了一次测试, 结果见表 4。表 4 与表 2 对比可知, 当 CPU 数大于 2 时, 所得的并行加速比较相应的 OpenMP

表 2 无嵌套模式 ($73 \times 76 \times 23$) 积分 1 天的 OpenMP 并行效能

CPU 数	1	2	4	6	8
用户时间 / s	3765.56	3700.34	3923.38	4256.01	4398.58
系统时间 / s	2.00	2.12	2.63	3.01	3.44
运行时间 / s	3784.43	1940.7	1063.0	714.0	570.8
并行加速比	1.0	1.95	3.56	5.30	6.63

表 3 嵌套模式 ($73 \times 76 \times 23$ 和 $76 \times 79 \times 23$) 积分 1 天的 OpenMP 并行效能

CPU 数	1	2	4	6	8
用户时间 / s	16398.42	16403.58	16825.48	17493.85	17853.93
系统时间 / s	8.24	10.95	8.30	9.76	8.02
运行时间 / s	16542.64	8312.88	4230.85	2867.00	2288.05
并行加速比	1.0	1.99	3.91	5.77	7.23

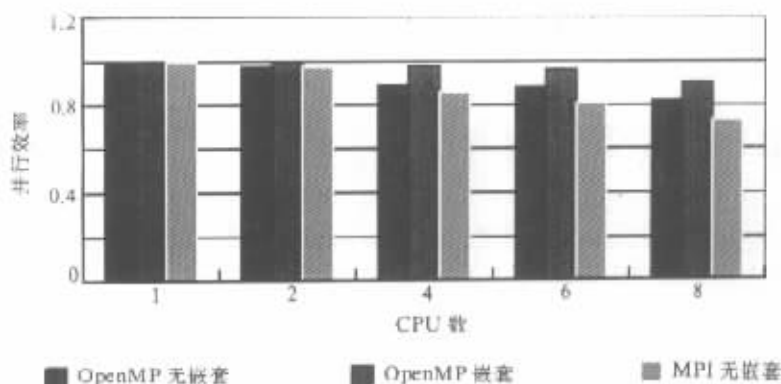


图 2 MM5 在 SGI Origin 2000 上的并行效率

并行方式所得的加速比均小。图 2 所示的并行效率也要低。对一个个例而言, 我们不能下结论认为: OpenMP 的并行效果比 MPI 的好。但起码可以得到结论说: 较好地运用 OpenMP 并行编程技术能达到较高的并行加速比。表 2、4 显示, OpenMP 程序在单 CPU 上运行比相应的 MPI 程序快 10% 左右。经初步分析, 我们推测这大概是因为 OpenMP 应用的是多线程 (Multi-threads) fork-join 模式, 而 MPI 应用的是多进程 (Multi-process) 模式, 在并行化较好的情况下, 前者比后者并行开销要小。

表 4 无嵌套模式 (73×76×23) 积分 1 天的 MPI 并行效能

CPU 数	1	2	4	6	8
用户时间 / s	4130.64	4077.2	5028.34	5434.52	5635.63
系统时间 / s	5.01	5.45	7.25	8.21	9.65
运行时间 / s	4350.42	2230.98	1264.66	895.15	744.93
并行加速比	1.0	1.95	3.44	4.86	5.84

最后, 我们可以根据表 2、3、4 的数据对 MM5 的可并行性作一粗略的分析。设在单 CPU 时, 运行一个程序需要时间为 t_1 , 其中计算程序中只能串行执行部分的时间为 s , 所以, 计算可并行执行部分的时间为 $t_1 - s$ 。在理想状况下 (即不计并行开销, 不计系统开销, 保证独占 CPU), p 个 CPU 参与计算所需时间 t_p 应满足以下关系:

$$\frac{t_1 - s}{p} + s = t_p. \quad (4)$$

求出 s , 则程序中可并行部分运算量占总运算量的比例为 $(t_1 - s) / t_1$ 。

根据表 2 中的数据, 由 (5) 式我们求得无嵌套设置的可并行部分比例为 97.4%。在理想的情况下, 考虑到有并行开销, 可以粗略地估计 MM5 的可并行部分计算量占总计算量的 95% 以上。按 Amdahl 定律, 无嵌套设置 (73×76×23, 60 km) 的并行加速比在理论上最大可达 20。同样, 由表 3 所计算的结果则都在 97% 以上, 所以, 两重嵌套设置 (73×76×23, 60 km; 76×79×23, 20 km) 的理论并行加速比最大可达 30。我们在分析了程序的结构以后也发现, 模式中除输入输出及一些模式初始化处理必须串行处理外, 进入预报循环后, 基本上都是并行处理的。而串行处理部分所需的时间变化不大, 随着运算量的增加和预报时间的延长, 串行处理部分所占的时间比例也越来越小。所以, 以上的分析与实际情况是吻合的。

5 总结

综上所述, 我们可以得到以下几点结论与思考:

(1) OpenMP 并行编程标准简单。对非并行用户而言, 并行指导语句对原程序无影响, 总共 50 000 多行的程序中, 只有 100 多行的 OpenMP 的指导语句。并且这些并行指导语句对原有的物理参数化过程, 动力框架和输入输出没有任何影响。所以, OpenMP 适于将已经存在的大量气象模式并行化, 基本上不必对程序做较大改动。

(2) MM5 的 OpenMP 版本的并行测试数据结果是很令人满意的。对双重嵌套运行的测试, 在 8 个 CPU 的情况下, 并行加速比达到了 7.23, 并行效率也超过了 0.9。

这一方面说明, OpenMP 并行编程标准确实有效可行; 另一方面也说明, MM5 模式的程序设计结构是合理的, 为成功地并行化奠定了基础。

(3) 个例研究表明, 用 OpenMP 并行版本运行所得的并行加速比优于用 MPI 并行版本所得的加速比。结果说明, 尽管 OpenMP 并行编程技术较之 MPI 简单, 但也能达到较高的并行加速比。

(4) 对于气象中尺度数值模拟和预报来说, 为了提高预报准确度和预报时效, 发展并行模式是一条切实可行的路子。特别是 OpenMP 较之气象界常用的 MPI 编程标准简单易学, 更适合于业务和研究人员使用。

参 考 文 献

- 1 Haagenson, P. L., J. Dudhia, D. R. Stauffer, G. A. Grell, The PennState / NCAR Mesoscale Model (MM5) Source Code Documentation, Tech. Rep. NCAR / TN-392+STR, National Center for Atmospheric Research, Boulder, Colorado, 1994.
- 2 Grell, G. A., J. Dudhia, D. R. Stauffer, A Description of the Fifth-Generation Penn State / NCAR Mesoscale Model (MM5) Tech. Rep. NCAR / TN-398+STR, National Center for Atmospheric Research, Boulder, Colorado, 1994.
- 3 PSU / NCAR Mesoscale Modeling System Tutorial Class Notes and User's Guides: MM5 Modeling System Version 2, June 1998, National Center for Atmospheric Research, Boulder, Colorado.
- 4 John Michalakes, The Same-Source Parallel MM5, <http://www.mmm.ucar.edu/mm5>.
- 5 John Michalakes et al., The Distributed-Memory Parallel Option to MM5, <http://www.mmm.ucar.edu/mm5>.

Some Study on Application of OpenMP in Mesoscale Meteorological Model—MM5

Zhang Xin, Ji Zhongzhen and Wang Bin

(State Key Laboratory of Numerical Modeling Atmospheric Sciences and Geophysical Fluid Dynamics,
Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029)

Abstract A new parallel programming standard—OpenMP is introduced in the beginning of this article. Its advantage and disadvantage are known through comparison with some other parallel programming standards. The application of OpenMP in MM5 of PSU / NCAR is presented in detail. The parallel performance of both OpenMP and MPI have been tested on SGI Origin 2000. The result show that OpenMP is a simple and efficient parallel programming standard and especially suitable for meteorological sciences field.

Key words: OpenMP; parallel computation; mesoscale model; parallel speedup; parallel performance