

用 OpenMP 并行化气象预报模式试验*

朱政慧 施培量

(国家气象中心,北京 100081)

颜 宏

(中国气象局,北京 100081)

提 要

介绍可移植的共享内存并行编程应用程序接口 OpenMP,通过对浅水波模式和复杂地形下的嵌套细网格模式的并行计算试验,探讨它用于并行化气象预报模式的可行性,并将其与 MPI 并行编程接口作比较。

关键词: 并行 OpenMP MPI

引 言

在并行计算领域,尽管采用共享内存编程方式具有简单易行的优点,但长期以来可移植性较差一直阻碍着开发者们在并行化其应用程序时采用这种方式,另一方面,传统的分布式内存并行采用可移植的消息传递方式,如 MPI 或 PVM,但是这种方式难以编程。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(API)的工业标准建议——OpenMP。目前,它已得到了包括 SGI,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。

在并行机体系结构方面,近年来主要厂商的高端并行机产品在单结点内都纷纷采用多处理器共享内存的技术,分布共享的理论也日趋成熟。在这种情况下,在深入研究基于消息传递的大规模并行应用软件的设计方法的同时,开展以 OpenMP 为基础的可移植可扩展共享内存式并行应用软件的设计方法的探索和研究,不但是可能的,也是必需的^[1]。

MPI 作为最流行的消息传递编程接口,主要应用于分布式内存并行计算机平台,但也可在共享内存并行机上使用。但是,由于该并行方式需要明确地划分数据结构并重构源程序,编程比较难。而 OpenMP 则应用于共享内存的并行计算平台,它是一组编译指导语句和可调用运行时库函数的集合,可被直接嵌入源程序而无需作较大的修改,编程简单,为任一现有软件的并行转换提供了一条渐进路径。总的来说,二者都是很重要的并行处理手段。

气象部门一直以来都是高性能计算应用的一个重要领域。大量的气象预报模式也急

* 本文由国家重点基础研究发展规划项目(G1998040910)资助。

2000-07-19 收到,2000-11-08 收到再改稿。

需并行化,尽管目前向量并行算法和以 MPI 为主要并行软件标准的消息传递编程方式的应用已较为广泛和成功,但仍满足不了天气和气候预报的需求。开展用 Open MP 并行化气象预报模式的探讨,不仅是当前高性能计算领域软硬件发展趋势的要求,也是气象预报业务发展的要求。在本文中,我们着重探讨了 Open MP 这种可移植的共享内存并行编程方法,并通过几个对比试验探讨其在气象应用中的可行性及与 MPI 并行方式的比较。

1 并行试验的基本方案

我们在 SGI Origin 2000(一种典型的分布共享型并行计算机,以下简称 O2000)上进行了全部的试验测试。分别对浅水波模式(以下简称 SW 模式)和复杂地形下的嵌套细网格模式(以下简称 YH 模式)进行了并行化处理,对模式的详细介绍请参见文献[2]与[3]。

试验方案如表 1:

表 1 并行试验方案

API	SW 模式		YH 模式	
	576 × 144	1536 × 384	69 × 38	273 × 146
Open MP	试验 1		试验 3	
MPI	试验 2		试验 4	

分别测试了两个模式的两种计算格点量,即对 SW 模式水平格点数($I \times J$)为 576×144 和 1536×384 , YH 模式则为 69×38 和 273×146 。

Open MP 的并行版本是在充分优化串行模式后的基础上开发的。依据对模式各部分耗时的统计,我们依次将其并行。在判断不存在数据相关的部分后,我们可以在粒度(granularity)较大的循环外加入并行指导语句,即 I, J, K 方向均可分给多个 CPU 运行。下面可通过两个例子来说明:

```
C$ OMP PARALLEL DO DEFAULT(SHARED)
```

```
C$ OMP & PRIVATE(J,I,XPX)
```

```
C$ OMP & REDUCTION(+ :EST)
```

```
DO 121 J=1, NLAQ
```

```
DO 121 I=1, NLOQ
```

```
IF(.NOT.KL) XPX = XDP
```

```
IF(KL) XPX = PIQ(I,J)
```

```
VORT(I,J) = (FGEO(J) + VORT(I,J)) / XPX
```

```
IF(MOD(MN,2).NE.0.OR.MK.NE.0)
```

```
& EST = EST + 0.5 * DSIGLA(KLA) * VORT(I,J) * VORT(I,J) * XPX /
```

```
& (PIQ(I,J) + DPU + DPB)
```

```
121 CONTINUE
```

这段程序用来求总的位涡度拟能值 EST。Open MP 指导语句添加在循环语句的最外层,程序首先作为一个单进程执行,碰到 PARALLEL DO 结构时,包含一个或多个进程的过程组就形成了,同时产生每个组员的数据环境。由于大部分变量是被所有的进程共享

访问的,所以设置缺省数据域为 SHARED;而并行区中所有对 J,I 和 XPX 的访问都是专用的,非共享的,应声明为 PRIVATE 变量;EST 具有 REDUCTION 属性,这样在程序结尾处就可对所有的专用拷贝归约求和,使主拷贝(master)得到正确值。循环结束处是一个隐式的路障,并行区在此终结。

下面再看一个较复杂的例子:

```
COMMON / WORK4/ VP( KLAX) , VY( KLAX) , VZ( KLAX) , VT( KLAX) , VQ( KLAX)
...
C$ OMP  PARALLEL DO SHARED( DEFAULT)
C$ OMP & PRIVATE( J, I, KLA, VT, VP, VY, VZ)
      DO 300 J = 1 , NLAP
      DO 300 I = 1 , NLOP
      DO 301 KLA = 1 , KLAX
      VT( KLA) = TBAR( I, J, KLA)
301    VP( KLA) = SGMTOP( PSBAR( I, J) , SIGLA( KLA) )
      CALL GPLG( VP, VY, KLAX)
      CALL GGAMA( G( 1, I, J) , VT, VY, KLAX)
      CALL GZLWST( PSBAR( I, J) , ZS( I, J) , VT, VY, G( 1, I, J) , VZ, KLAX, 0)
      DO 302 KLA = 1 , KLAX
302    Z( KLA, I, J) = VZ( KLA)
300    CONTINUE
```

这段程序包含了过程调用, TBAR 和 SGMTOP 是两个函数。由于 VT, VP, VY 和 VZ 定义在公共块中,其缺省属性为 SHARED,但此处对它们的访问实际上是各进程对各自的专用拷贝进行读写,是非共享的,显然应声明为 PRIVATE 变量。这种潜在的矛盾有两种解决办法:一是将它们改声明为局地工作数组,二是在程序中或编译时说明此公共块为专用变量。气象预报模式中各子程序间存在大量的数据要通过公共块传递,因此这一问题需要特别注意。

SW 和 YH 模式的 MPI 并行版本均采用了静态水平区域分割方法,即将水平(I, J) 方向上的计算尽可能平均地分配给多 CPU,并在程序中加入适当的通信,让程序并行执行。鉴于这种并行化方法已大量成功地运用于气象模式中,这里就不作详细介绍了。

2 SW 和 YH 模式的并行试验结果分析

2.1 OpenMP 的可行性试验

下面是用 OpenMP 并行 SW 模式和 YH 模式后所得的试验数据。分别选用了两种计算格点量进行计算。表中运行时间为在 O2000 上运转的墙钟时间,单位为秒,下同。

(1) SW 模式 表 2 和图 1 分别为用 OpenMp 并行化 SW 模式的运行结果及并行效率。

表 2 用 OpenMP 并行化的 SW 模式运行结果

CPU 数	576 × 144		1536 × 384	
	运行时间(s)	加速比	运行时间(s)	加速比
1	386.99	-	11234.09	-
2	205.64	1.88	5761.78	1.88
4	116.58	3.33	3431.16	3.27
8	71.93	5.44	1944.73	5.78
16	49.27	7.85	1313.35	8.53

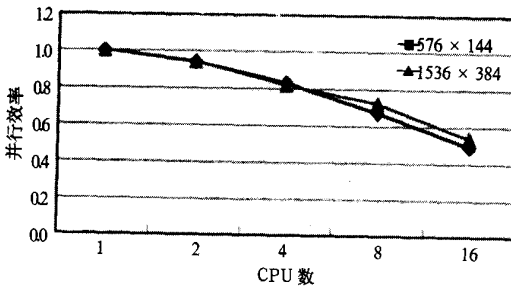


图 1 OpenMP 并行 SW 模式并行效率

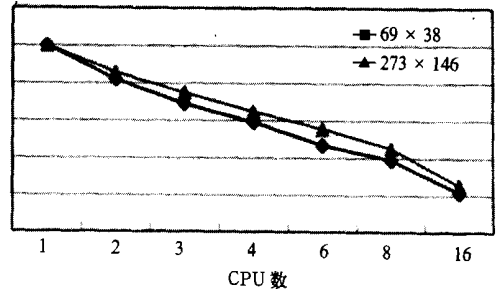


图 2 OpenMP 并行 YH 模式并行效率

(2) YH 模式 表 3 和图 2 分别为用 OpenMP 并行化 YH 模式的运行结果及并行效率。

表 3 用 OpenMP 并行化的 YH 模式运行结果

CPU 数	69 × 38		273 × 146	
	运行时间(s)	加速比	运行时间(s)	加速比
1	202.62	-	14069.64	-
2	123.73	1.64	8220.66	1.71
3	97.86	2.07	6281.17	2.24
4	85.85	2.38	5421.27	2.60
6	71.97	2.82	4214.46	3.34
8	64.74	3.16	3887.97	3.62
16	59.29	3.42	3462.63	4.06

从表 2、表 3 和图 1、图 2 可以看出,随着计算格点量的增大,加速比和并行效率明显提高。这是存在相对并行开销的原因。这种并行开销的时间要花在产生子进程、开始和结束并行区,在并行区结束时进行同步以及为并行化所加的额外代码的执行等方面。当计算格点量增大时,这种开销所花费的时间相对于自身计算所花费的时间越来越小,因而加速比和并行效率就会明显提高。鉴此,我们在用这种方式并行化串程序时,就应该尽量并行化那些计算量较大的循环,在加指导语句时可以配合使用条件判断语句来选择并行计算量合适的循环。另外,要根据实际程序所需而指定参加计算的 CPU 数。过分地要求增加参加并行运算的 CPU 数,计算效率下降,反而造成资源浪费。如对 YH 模式来说,用 6 个 CPU 进行计算就行了。

在计算量固定不变的条件下,随着参与计算的 CPU 数增多,并行效率降低。这点正

是由 Amdahl 定律所阐明的规律:程序中存在的串行执行部分严重地阻碍了并行效率的提高。因此对于共享内存并行编程的开发者来说,改动源程序,消除数据相关,以尽量减少程序中的串行执行部分是一项重要任务。有利的是它是一种渐进并行的方式,而这种改动又只是很小的一部分。

2.2 MPI 并行版本运行结果

下面是 SW 和 YH 的测试结果:

(1) SW 模式 表 4 和图 3 分别为用 MPI 并行化 SW 模式的运行结果及并行效率。

表 4 用 MPI 并行化的 SW 模式运行结果

CPU 数	576 × 144		1536 × 384	
	运行时间(s)	加速比	运行时间(s)	加速比
1	342.34	-	10242.98	-
2	174.32	1.97	4990.16	2.05
4	89.19	3.84	2450.50	4.18
8	52.80	6.58	1235.17	8.29
16	28.47	12.02	655.95	13.10

(2) YH 模式 表 5 和图 4 分别为用 MPI 并行化 YH 模式的运行结果及并行效率。

表 5 用 MPI 并行化的 YH 模式运行结果

CPU 数	69 × 38		273 × 146	
	运行时间(s)	加速比	运行时间(s)	加速比
1	223.73	-	15218.49	-
2	116.87	1.91	7718.84	1.97
3	81.97	2.73	5205.04	2.92
6	-	-	2588.21	5.88
10	-	-	1607.26	9.47
11	-	-	-	-

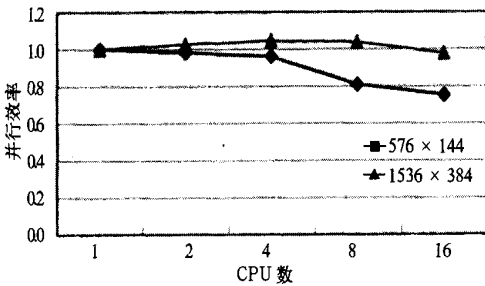


图 3 MPI 并行 SW 模式并行效率

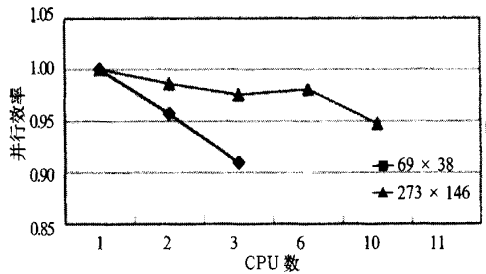


图 4 MPI 并行 YH 模式并行效率

从表 4、表 5 和图 3、图 4 中可以看出,当计算量达到一定大小时,并行效率能大于 1,也就是出现了所谓的超线性现象(superlinear)。这是由于计算机存在多层存储体系造成的。当计算量较大时,数据在外存、内存、缓存和寄存器之间移入移出,需要花费时间。多个 CPU 并行执行时,每个 CPU 的计算量减小,数据移动时间减小,若其减小程度比花在结点间的通讯时间要大,就会出现超线性现象。当参与计算的 CPU 数足够多,数据移动

时间可忽略不计,由于结点间的通讯开销增大,并行效率下降。但总的来说,仍是计算量越大,并行效果越好。

2.3 OpenMP 和 MPI 的比较

针对不同的计算机平台和应用程序,OpenMP 和 MPI 都是不可缺少的重要的并行编程应用程序接口。它们都具有较好的可移植性和可扩展性,并提供较好的性能。在并行化气象预报模式时,采用何种方式都是可行的。

从编程角度考虑,OpenMP 明显地比 MPI 容易被接受,开发并行模式的时间较短,也很灵活,在不存在数据相关的情况下,OpenMP 的并行可以从水平或垂直方向进行,而对于 MPI 来说,由于垂直方向上的数据相关性太强,因此并行区域的划分适于在水平方向上进行。从并行粒度上来说,OpenMP 着重于循环级、过程级的开发, MPI 则是更粗粒度的程序级并行,因而后者更易获得较高的并行效率。从试验结果中也可看出,OpenMP 版本的并行模式效率明显低于 MPI。另外,当 CPU 数比较多时,OpenMP 所提供的性能远不如 MPI。

3 结论与讨论

无论采用何种编程方式,并行效率都与计算量,特别是可并行计算量相对于并行开销的大小有关,计算量越大,并行效率越高;计算量大小一定时,CPU 数越多,效率越低。具体到气象预报模式上,则是计算格点数越多,并行效率越高。对于用 MPI 并行化的模式,内点计算量相对数据交换量越大,效率越高。而对于用 OpenMP 并行化的模式,则是并行执行部分相对于串行执行部分越大,效率越高。总的来说,利用 OpenMP 并行化气象预报模式是可行的、有效的,同时还具有简单明了、灵活性强的优点。

需要指出的是,虽然用 OpenMP 编程是可行的,但其并行效率还不够理想,原因是多方面的。首先是因为程序中还存在许多串行执行部分。因此,最大限度地并行化源程序,是获得理想的 OpenMP 并行程序并行效率的前提。其次,因为 OpenMP 的并行粒度比 MPI 的要小。另外,同 MPI 应用程序接口相比较,OpenMP 还不能算作一个很成熟的产品,我们这里所用的 OpenMP Fortran 应用程序接口还只是 1.0 版,其性能还有待于进一步提高。因此,我们将继续关注今后其更高版本的使用效果。

另外,需要进一步深入研究和探讨的就是并行和 Cache 的关系问题,这里不作详细说明。尽管我们已经认识到加入不当的并行指导语句后,由于 Cache 的一致性问题,会出现某些变量的假共享,致使程序运行速度变慢,但如何有效地识别和修正这种错误还有待研究,这也是一个很有意义的研究内容。

参考文献

- 1 施培量,朱政慧.可移植共享内存并行编程 API——OpenMP.国家气象中心科技年报,1998.
- 2 颜宏.复杂地形条件下嵌套细网格模式的设计(一)数值模式的基本原理.高原气象,1987,6(2)(增刊):1~62.
- 3 颜宏.复杂地形条件下嵌套细网格模式的设计(二)次网格物理过程参数化.高原气象,1987,6(2)(增刊):63~138.

**STUDY ON THE PARALLELIZATION OF METEOROLOGICAL
FORECASTING MODELS WITH OPENMP AND
COMPARISON TO MPI METHOD**

Zhu Zhenghui Shi Peiliang

(*National Meteorological Center , Beijing 100081*)

Yan Hong

(*China Meteorology Administration , Beijing 100081*)

Abstract

A portable shared memory parallel application program interface (API) OpenMP is introduced. By conducting parallel experiments on a shallow water model and a nested fine-mesh model over the complex topography, its feasibility for the parallelization of meteorological forecasting models is discussed with comparison to Message Passing Interface (MPI) .

Key words : Parallelization OpenMP MPI