

海气耦合模式的优化方法研究*

魏 敏¹⁾²⁾ 罗 勇³⁾ 王兰宁³⁾ 董 敏³⁾ 李清泉³⁾ 徐 影³⁾

¹⁾(北京大学物理学院,北京 100871) ²⁾(国家气象信息中心,北京 100081)

³⁾(中国气象局 气候研究开放实验室,北京 100081)

摘 要

气候模式不断发展的同时,对高性能计算机提出了更高的要求,如何提高模式在现有计算机资源上的运行效率问题已越来越重要。文章介绍了利用编译器优化和手工优化技术对海气耦合模式进行优化试验,优化后模式运行效率提高 60%,表明将这些优化方法应用到气候模式的改进工作中,可以达到较好的效果。

关键词:高性能计算 气候模式 优化方法

引 言

气候模式是自然科学领域中应用最复杂、最精细的计算机模式之一,对计算资源有巨大的需求。全球海气耦合模式的大气部分最初来源于 T63 中期预报模式^[1],目前影响该模式运行效率的问题主要有两方面:一方面,由于其物理过程复杂,包含大量的科学运算,对计算资源的需求很大;另一方面,经过再次开发、移植后,部分程序结构不能充分利用现有 Memory(内存)、Cache(高速缓存)等硬件的优良特性。目前,国家气候中心使用全球海气耦合模式进行 8 个样本的集合预报,样本个数与运行平台 CPU 个数相同。我们对该模式进行串行优化,不仅满足了现阶段业务需求的时效性,也可以为将来模式的并行化打下良好的基础。

1 海气耦合模式运行平台简介

本次模式调试是在 IBM 公司 P670 服务器(集合预报系统运行平台)上进行的,P670 服务器有 8 颗 1100 MHz POWER4 CPU,16 GB Memory。POWER4 芯片采用 MCM(多芯片模块结构,Multi-Chip Module),有 L1、L2、L3 三级 Cache。每颗 CPU 包含一个 64 KB L1 指令 Cache 和一个 32 KB L1 数据 Cache;每个芯片包含 2 颗 CPU,共享 1.44 MB 的 L2 Cache;每个模块包含 4 个芯片,共有 2×4 颗 CPU,共享 128 MB 的 L3 Cache。

2 海气耦合模式优化方法和过程

对程序进行优化的过程中,采用了编译器优化和手工优化两种方法。采用编译器特

* 中国气象局气候变化专项 LC2003C-1 资助。

2004-06-17 收到,2004-12-08 收到再改稿。

定的编译选项,在编译的时候将程序的源代码尽可能的自动进行优化,可以在很大程度上改善模式运行状况。这里我们主要介绍的是人工参与较多的人工优化方法。这种方法在一定程度上可以为编译器优化创造条件,即先将部分语句手工修改成利于编译器识别的形式,保证最大程度地利用编译器自身优点。同时,手工调优方法也可以针对程序总体框架和部分数据结构进行,这是编译器优化方法不能实现的。模式优化涉及到程序本身结构的调整和使其充分发挥运行平台软硬件资源特性两个方面。由于模式中大气部分的运算占了 80% 以上的计算量,因此针对大气模式部分进行优化,对提高模式整体运行效率可以起到立竿见影的效果。

在模式优化之前,我们采用 gprof 分析工具确定程序的“hot point”,即程序中哪部分是 CPU 密集型代码。表 1 是模式优化之前积分 1 个月所得到的运行效率报告的部分内容,由“调用次数”和“执行时间”可以看到程序调用 `_exp`, `_log` 数学运算的次数非常多,占用了大量的 CPU 时间。如果单个函数执行大多数工作的话,优化、加速该函数可能会使程序运行速度显著加快;另一方面,如果该函数极少调用并且无论如何都不会占用很长时间,则对它的优化可能对总体性能不会有太大的改进。因此我们应该首先对这些占用 CPU 时间最多的函数进行优化。

表 1 模式优化之前积分 1 个月运行效率报告的部分内容

函数名称	占用程序执行 时间百分比/%	执行 时间/s	累计执行 时间/s	调用次数
<code>_exp</code>	22.0	3068.83	3068.83	20438206848
<code>qcfel</code>	9.8	1372.74	4441.57	61991935
<code>_mcount</code>	9.1	1266.09	5707.66	
<code>_log</code>	5.5	768.79	6476.45	5975941760
<code>comscp_rm</code>	3.9	550.51	7026.96	138834432

2.1 充分利用系统的硬件特性

通常的计算可以分为两部分:处理器实际计算的时间和对计算所需数据存取的时间,其中后者占有相当大的比例,也是我们可以设法改进的。Cache 的使用缓解了 CPU 和 Memory 之间速度差距过大的矛盾。如图 1 所示,当 CPU 在运算中访问某一数据时,首先在 Cache 中查找,如果在 Cache 中找到该数据,称为命中;如果没有找到,则从 Memory 中查找该数据装载到 Cache 中。Cache 和 Memory 交换数据是以“Cache line”为单位进行的。如果“Cache line”的大小是 128 字节,则 CPU 每次从 Memory 装载连续存放的 128 字节数据到 Cache 中,即 CPU 访问

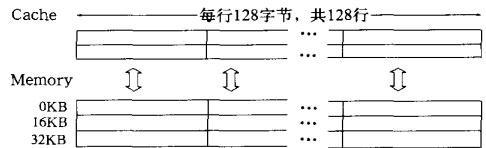


图 1 Cache 与 Memory 间数据交换示意图

某一个数据时,这个数据和与之相邻的共 128 字节数据会一起被装载到 Cache 中。我们可以对 Memory 中存放数据的顺序进行调整,保证一次装载操作取到尽可能多的有用数

据^①,提高 Cache 命中率。

数据在 Memory 中以线性方式存储。stride(跨距)指连续的数据元素存储在 Memory 中的间隔,如 stride = 1,表示顺序地存取 Memory 中的连续数据;stride = n,表示连续存取的数据在 Memory 中存放的位置相隔 n 个元素。从上面关于 Cache 特性的介绍,我们可以知道 stride = 1 时,可以最有效地利用 Cache 行。Fortran 语言的数组元素按列优先顺序存储。所有与经圈 I,纬圈 J,层次 K 相关的三维数组结构都是按照 a(i, j, k) 排列,应保证数组索引的顺序和循环顺序一致,如图 2 所示,使 Cache 中数据具有空间的连续性。

```

do k = 1, Kmax
  do j = 1, Jmax
    do i = 1, Imax
      a(i, j, k) ...
    end do
  end do
end do

```

图 2 循环顺序示意图

2.2 充分利用高效数学函数库

在气候模式的各物理过程中,包含大量的科学运算。操作系统标准的数学函数库中一些复杂的运算,如除法运算将占用 20 ~ 24 个 CPU 周期,增加 CPU 的开销。IBM 公司在 AIX 系统环境中,提供了 Mathematical Acceleration Subsystem (MASS) 函数库^[2]。MASS 函数库分为标量和矢量两种,可以在应用程序链接时指定^②。如果使用标量 MASS 函数库,不用修改源程序,程序运行过程中,会自动调用 MASS 函数库中优化过的函数而不再调用标准函数。矢量 MASS 函数库中的函数可以将多个相同类型的数学运算,作为一个矢量运算一次完成,前提是需要对源程序代码进行一定的修改。使用矢量函数库,可以获得更好的效果(表 2)。

表 2 标准函数与 MASS 函数使用 CPU 周期个数及其加速比

	标准函数使用 CPU 周期个数 (ST)	标量 MASS 使用 CPU 周期个数 (SM)	加速比 (ST/SM)	矢量 MASS 使用 CPU 周期个数 (VM)	加速比 (ST/VM)
SQRT	125	72	1.74	7	17.85
EXP	89	42	2.12	14	6.36
LOG	167	84	1.94	15	11.13
X**Y	405	183	2.21		

引入矢量 MASS 库,需要分段改写原来的程序,使其能够符合调用矢量 MASS 函数库的语句特征。在表 3 的例子中,用 $b(i) = \exp(r * \log(z(i)))$ (127 个 CPU 周期) 替换了 $b(i) = z(i) ** r$ (183 个 CPU 周期),在进一步修改时,将 $\log(z(i))$ 的操作提到循环外部,调用矢量函数 vslog(),一次完成了对 z 数组中 max 个元素的 $\log()$ 运算,结果保存在 tmp 数组中。减少了 $\log()$ 运算的次数,节省了代码运行时间。同理,可以完成对 $\exp()$ 运算的修改。

① Liao Jang Shieh. Program Optimization Considerations. Presentation, 2002.

② Christidis Zaphiris. Tuning Fortran Programs for Optimal Performance on RISC Superscalar Processors. 2003.

表 3 使用矢量 MASS 函数库修改程序代码例子

原始程序	初步修改	进一步修改
do i = 1 , max b(i) = z(i) ** r a(i) = b(i) * dz end do	do i = 1 , max b(i) = exp(r * log(z(i))) a(i) = b(i) * dz end do	call vslog(tmp , z , max) do i = 1 , max tmp(i) = tmp(i) * r end do call vsexp(b , tmp , max) do i = 1 , max a(i) = b(i) * dz end do

2.3 语句优化

- (1) 将公共子表达式放在括号内,利于编译器识别。
- (2) 将循环内的常量计算提到循环外,避免不必要的重复计算。
- (3) 尽量减少在循环中使用 if 语句。
- (4) 合并小规模循环可以有效地减少系统开销^[3]。

2.4 调整模式的输出方式

过去由于受计算机系统内存容量的限制,模式输出部分采用多次 I/O 写盘方式,导致程序运行速度受到 I/O 的限制(I/O bound)。目前模式运行平台的内存容量比以往有了显著的提高,因此修改了有关数据输出部分的程序,采用以“空间换时间”的策略,用缓冲写的方式,在内存中开辟了一个较大的缓冲区,将数据暂时存放,到一定程度后,一次性写盘,这样减少写盘次数,节省模式 I/O 时间。

3 海气耦合模式优化结果

对模式部分子程序进行修改后,使模式的运行效率有一定提高。在没有进行优化之前,积分 1 个月需要 28545.94 s,优化之后为 8611.23 s,运行效率提高了近 60%。我们对优化前和优化后的模式分别做了 10 年积分试验,取得一致结果,证明优化过程没有影响模式运行的准确性。表 4 是模式经过各种方法优化后积分 1 个月所用时间的统计,每一项是在前一项基础上进行的,所得结果反映的是前面各项的总体效应。

表 4 模式积分 1 个月运行时间统计

优化方法	未优化	调整输出文件	采用编译器优化选项	调整语句 使用 MASS 函数库
时间/s	28545.94	28213.40	11908.04	8611.23

4 结 语

根据以上的试验结果,我们可以认识到,通过应用编译器调优和手工调优等成熟的优化技术,可以在一定程度上提高模式的运行效率,缓解计算资源紧缺的矛盾,这将是促进气候模式发展的有效方法之一。我们在今后的工作中要进一步开展对气候模式优化方法的研究,使其发挥出更大的作用。

参考文献

- 1 董敏.国家气候中心大气环流模式——基本原理和使用说明.北京:气象出版社,2001.
- 2 <http://techsupport.services.ibm.com/server/mass>. Mathematical Acceleration Subsystem for AIX.
- 3 <http://www.redbooks.ibm.com>. IBM.Optimization and Tuning Guide for Fortran, C, and C++.1993.

A STUDY ON THE OPTIMUM ALGORITHM OF GLOBAL OCEAN ATMOSPHERE COUPLED MODELS

Wei Min¹⁾²⁾ Luo Yong³⁾ Wang Lanning³⁾ Dong Min³⁾ Li Qingquan³⁾ Xu Ying³⁾

¹⁾ (School of Physics, Peking University, Beijing 100871)

²⁾ (National Meteorological Information Center, Beijing 100081)

³⁾ (Laboratory for Climate Study, CMA, Beijing 100081)

Abstract

The continuous development of climate model proposes higher requirements to better performance of computers. How to improve the capability of climate model running on current-stage computer resources becomes more and more urgent. The article is devoted to the optimization of Ocean-atmosphere Coupled Model via an editor or manually. It is shown that the optimized Global Ocean-atmosphere Coupled Model can raise the efficiency of the model operation by 60%, indicating that the optimization can help improve the performance of the model.

Key words: High performance computing Climate model Optimum algorithm