软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, 2011,22(Suppl.(2)):192−198

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# MPMD 程序 CCSM3 的负载均衡分析*

吴 宏[1+], 翟 琰[2], 翟季冬[2]

[1](江南计算技术研究所,江苏 无锡 214083)

[2](清华大学 计算机科学与技术系,北京 100084)

## Load-Balance Analysis for the MPMD Program CCSM3

WU Hong[1+], ZHAI Yan[2], ZHAI Ji-Dong[2]

[1](Jiangnan Institute of Computing Technology, Wuxi 214083, China)

[2](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: E-mail: wx_wuhong@163.com

**Wu H, Zhai Y, Zhai JD. Load-Balance analysis for the MPMD program CCSM3. *Journal of Software*, 2011,22(Suppl.(2)):192−198.** http://www.jos.org.cn/1000-9825/11040.htm

**Abstract**: The model of MPMD program, which is based on MPI, is quite complex and includes several SPMD programs and their coupler. MPMD model is quite popular in climate domain, and it will be quite practical for the developer to understand its characters. This work focuses on the performance of the coupler module of MPMD program CCSM3.0 to locate the possible load-imbalance problem among the subprograms. The load balance issue of complex MPMD program is simplified down to issues of a set of SPMD programs and their interactions, providing good vision for the developers and performance diagnosing individuals to optimize the program.
**Key words**: MPMD; MPI; CCSM3; load-balance

**摘 要**: MPI 消息传递的 MPMD 并行计算模型非常复杂,通常由一组 SPMD 程序和耦合器组成.这种 MPMD 计算模型在气候科学计算中十分常见,因此有效的性能分析工具和方法对于开发人员具有非常实际的意义.以 MPMD 程序 CCSM3 为例,着重分析了 MPMD 程序与 SPMD 程序最显著的区别——耦合器上的性能事件,以耦合器为中心,去发现和定位不同的子程序之间的负载均衡问题,将复杂的 MPMD 程序的进程间关系简化为 SPMD 程序的交互及 SPMD 程序内部的负载均衡问题,从而帮助开发人员和性能调试人员更准确地发现程序中的负载不均衡现象,对程序的设置或者算法进行优化和改进.
**关键词**: MPMD;MPI;CCSM3;负载均衡

当今,重大挑战性科学难题,诸如人类基因、地球气候准确预报与模拟、核爆炸模拟等等,都与并行计算技术紧紧联系在一起.作为研究与计算工具,高性能并行计算机系统发展日新月异,2011 的 Top500 中已经出现了多台 P 级计算机[1].而作为并行计算的灵魂——并行算法与并行软件,可以对性能产生重大影响,但发展显得相对较缓.究其原因,首先并行算法的设计与并行软件的开发非常复杂,它不仅与应用领域本身有关,而且与并行计算机体系结构密切相关;其次,高性能并行计算机系统运算峰值越来越高,系统结构也随之越来越复杂,并行

* 基金项目: 国家自然科学基金(61103021); 国家高技术研究发展计划(863)(2010AA012403)
收稿时间: 2011-07-20; 定稿时间: 2011-12-01

应用课题如何高效地运行于高性能计算机系统也越来越不容易.因此,并行应用课题的开发、性能分析与优化亟需更深入的研究.

基于 MPI 消息传递的并行程序模型是当前最主流并行应用模式.根据从数据分解和功能分解两种划分方式,消息传递模型分为单程序多数据流模式(SPMD)和多程序多数据流(MPMD)[2].随着研究问题的越来越复杂,涉及的专业应用领域越来越多,MPMD 并行模型的划分粒度实际应用中会扩展到应用领域.举例来说,地球气候问题涉及众多科学领域:大气、海洋、陆地、海冰、人类活动、碳循环等等,不同专业领域的科学家在各自领域研究多年,积累了众多并行计算应用软件,一般为 SPMD 程序.气候问题必须综合各专业领域统一建模,将各个 SPMD 应用统一建模成相互作用的 MPMD 并行模型应运而生;各个领域 SPMD 应用通过耦合器发生相互作用,耦合器成为此类 MPMD 并行模型重要特征.

## 1 MPMD 并行计算特点

### 1.1 MPMD并行模型与负载均衡问题

SPMD 与 MPMD 同属于消息传递——主要为 MPI——并行模型.SPMD 程序各个进程同构,多个进程对不同的数据执行相同的代码;MPMD 程序各个进程异构,多个进程执行不同的代码,往往包含 SPMD 程序.在形式上可以用 SPMD 程序来伪造 MPMD,但是该并行程序本质上还是 MPMD 并行模型,MPMD 属性并不会改变.

SPMD 性能分析与优化,往往在于挖掘计算与通信的并行性,尽最大可能使得计算与通信重叠,其分析目标在于找到计算或通信的瓶颈,指导并行算法改进和程序优化,指导合适的并行规模.MPMD 并行计算性能分析不仅包括 SPMD 的性能分析,而且包括程序间的性能分析与调优.这包括程序间负载均衡问题、程序间通信与计算分析优化、并行模型瓶颈等等.

本文重点关注 MPMD 程序的负载均衡问题.现有的工作针对 SPMD 程序的负载均衡问题做了很多分 析[3−6].但 MPMD 程序的负载均衡问题有其自己的特点:耦合器的存在使得我们不需要对所有的进程进行整体的分析,通过分析耦合器进程上采集的性能数据,即可有效分析 MPMD 各个子程序的负载不均衡情况.

### 1.2 实例课题CCSM3

CCSM3[7]是具有重要现实意义的典型 MPMD 应用模式,本文以 CCSM3 为实例研究 MPMD 并行模型的负载均衡分析.CCSM3 应用模式主要用于研究与模拟全球气候变化,包括完全相互作用的大气、海洋、海冰和陆面 4 个分量模式,通过耦合器相互交换数据;各个模块相互独立并行运行,并周期性地与偶合器交换数据.CCSM3 并行结构如图 1 所示,各个分量模式通过耦合器相互作用,耦合器是整个 MPMD 程序结构的核心.

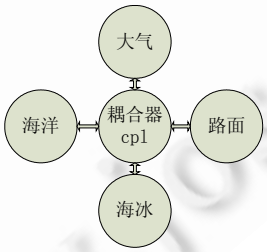



图 1 CCSM3 程序结构

## 2 MPMD 负载均衡分析:方法和工具

### 2.1 需要解决的问题

由于 MPMD 程序的子程序交互都是显式地通过耦合器进行,因此 MPMD 程序整体的负载均衡状况只需要观察耦合器的性能事件即可,这是 SPMD 程序所不能保证拥有的特性.而对于每个 SPMD 子程序的负载均衡的

分析,可以参考已有的方案,如文献[6]提出的统计方法.这样我们就将 MPMD 的负载均衡问题切分为若干子问题.本文重点关心的是如何解决 MPMD 子程序间的负载均衡问题.

### 2.2 性能数据的采集

当前主流的并行程序都是通过 MPI 编程接口实现.MPI 的规范中定义了 PMPI 的插装规范,因此用 MPI 接口实现通信插装非常容易.使用该接口的类似的性能工具包括 mpiP[8],Kojak[9]等.经实际测试,对于通信函数的简单插装并不会带来太大性能损失.本文用于分析负载均衡的数据目前均来自于我们自己实现的一套插装库.另外,由于我们更关心耦合器的性能开销,因此,对于耦合器进程,我们在必要时会采集详细的性能数据,如每一次的通信调用的时间.这种细粒度的采集对于大规模的并行程序是较为危险的,因为数据量非常大,但是只针对耦合器进程,在我们的实验中,是可以接受的.

### 2.3 通信子的分类

除了结构上的区别,MPMD 程序与 SPMD 程序在实现上的一个比较显著的区别就是 MPMD 程序存在更多数量的通信子.因此,按通信子对于通信进行分类会更有利于性能分析.由于 MPI 的实现不同,同一个通信子在不同的进程中对应的句柄可能是不相等的,因此我们需要截获新通信子创建事件,此时为一个同步点,然后利用 MPI_Bcast 将该通信子的 0 号进程的全局编号和通信子的值广播给其他进程,作为唯一标识,从而在最后输出时对结果进行合并.

## 3 实例:CCSM3 性能分析

### 3.1 实验环境

集群环境,单节点设置为 Intel Xeon X5670 处理器,每节点 48GB 内存,网络使用的 QDR Infiniband,存储系统为 NFS.软件环境方面,操作系统是 RHEL5.5,编译器使用的 icc/ifort 11.0.069,MPI 库为 Intel MPI 4.0.0.025.CCSM3 程序运行的进程数为 84 个进程,其中 0-7 为耦合器 cpl,8-15 为 csim,16-27 为 clm,28-51 为 pop,52-83 为 cam,程序运行的数据集为 T42_gx1v3,后文若未提及,均用子程序 1-5 来代替.

经过我们的分析发现,CCSM3 程序在给定的平台和输入下存在着非常明显的负载不均衡现象,子程序 4 直接影响了所有其他程序的完成时间.我们通过整体的性能结果和通信的统计结果确认了这一点.

### 3.2 整体性能结果

采集整体性能主要观察程序的通信和计算开销的比例.通过分析程序整体的负载均衡情况,可以发现比较明显的问题,再有针对性地进行深入分析.

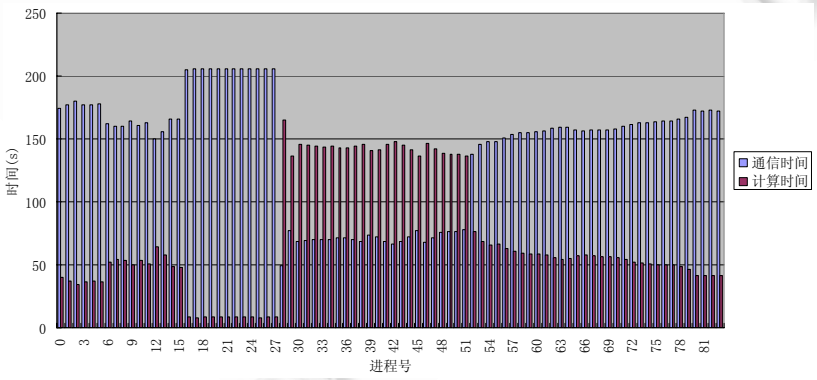



图 2 CCSM3 的整体计算通信性能

CCSM3 在实验环境中整体性能如图 2 所示,多次运行的总时间取总值为 214s,而对于除子程序 4 外的大部分进程而言,通信占总时间的比例都非常大.从整体的计算比例上看,整体的负载非常不均衡.只根据计算通信的比例尚无法断定究竟是由于负载不均衡还是通信量过大引起的此种现象,因此还必须进一步分析通信的具体开销.

### 3.3 各个通信函数的时间花费

通过统计通信函数的时间就可以看出比较明显的问题,从图 3 可以看出,MPI_Bcast 等组通信操作花费了大量的时间.另一方面,值得注意的是第 0,8,16 号进程,MPI_Recv 和 MPI_Waitall 操作占通信总时间的绝大部分比例.由于组通信需要等待通信子的所有进程都参与时才可以继续进行,因此对于前 3 个子程序,需要验证两件事情:(1) MPI_Bcast 的实际传输消息数量究竟有多少,是由于负载不均衡造成的长时间等待,还是由于通信的缓慢使得进程长时间陷入通信函数;(2) 在 MPI_Recv 上的巨大花费是什么造成的.最后一个子程序的时间主要花费在 MPI_Waitall 中,这种情况往往是由于使用了通信计算重叠的设计模式,而计算的数据量又较小引起的.随着进程数的扩展,在 MPI_Waitall 中的时间有望减少,而计算时间则会相应增加.对于这种通信模式,不属于本文的讨论范围,仅此提及.

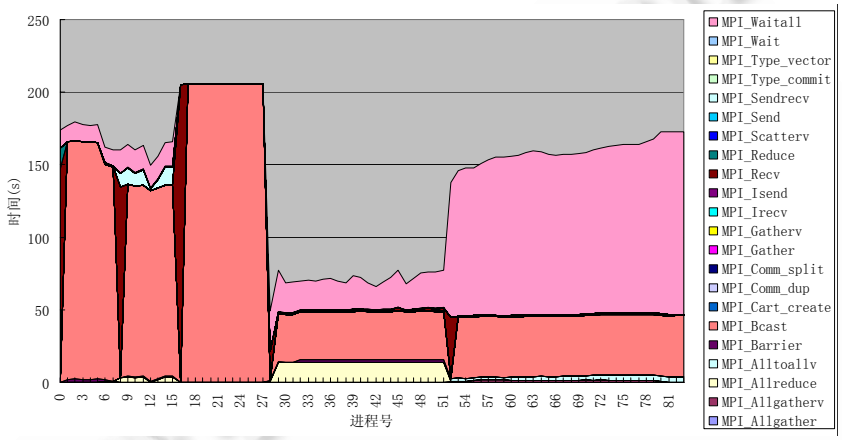



图 3 CCSM3 的整体通信性能

为了确定问题所在,我们采集了每个进程的 MPI_Bcast 发送的通信总量,以确定其是否可能成为影响的主要因素,如图 4 所示.

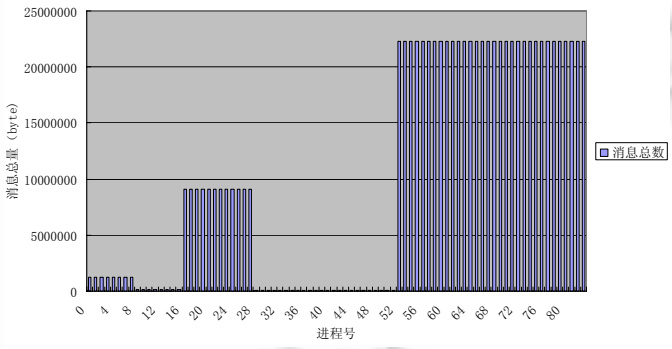



图 4 CCSM3 各个进程通过 MPI_Bcast 传递的消息数量总数

可以看出,在 MPI_BCast 上花费较多时间的进程实际上并没有发送较多信息,因此合理解释就是因为 MPI_Recv 而引起的通信缓慢.事实上,通过分通信子的统计,我们确认前 3 个子程序的组通信缓慢完全是由于 MPI_Recv 引起的,这是负载不均衡时所表现出的典型特征.

### 3.4 分通信子的通信性能统计

分通信子的统计模式可以更清楚地发掘各个进程之间的交互.CCSM3 在本文所进行的实验规模中创造了

17 个新通信子,包括默认的 MPI_COMM_WORLD,一共 18 个通信子(包括笛卡尔拓扑的通信子复制).限于篇幅,我们只列出几个主要的通信子加以分析.

从图 5 可以看出,所有花费时间较多的 MPI_Recv 均为全局通信子产生的,而且均发生在每个子程序的 0 号进程.由于第 2 节中讨论过,对于 MPMD 程序的负载均衡,我们收集耦合器的性能数据即可,因此我们细粒度地采集了 0 号进程的每次 Recv 的开销,发现了以下规律:

(1) 0 号进程的 Recv 操作仅作用于上面几个 Recv 开销较大的进程.

(2) 0 号进程在接收 28 号进程的数据时花费时间最大,仅仅 4 次 800 字节的 Recv 操作却陷入了通信函数长达 80s(总共 Recv 的花费为 150s 左右).

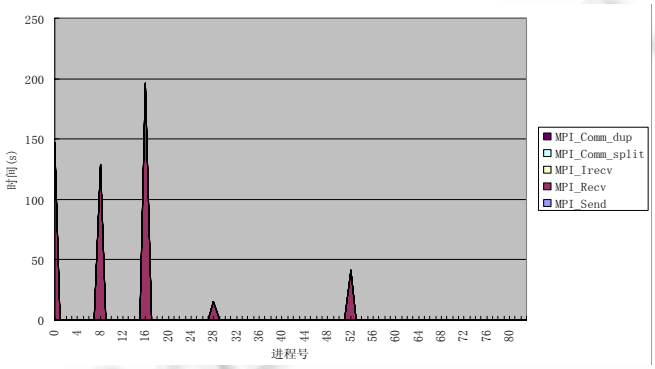



图 5 CCSM3 用于全局同步的通信子的通信统计

结合整体性能中子程序 4 整体较长的计算时间,不难确定,正是由于子程序 4 的负载过大,导致耦合器的等待,从而也延长了其余程序的执行时间.

更进一步地,通过通信子的划分,可以确定每个子程序其余进程巨大的 MPI_Bcast 开销均因为等待各自的 0 号进程,如图 6 所示.其余的子进程与此类似.至此,我们最终可以确认子进程 4 的负载不均衡现象和各种通信开销巨大的具体原因.

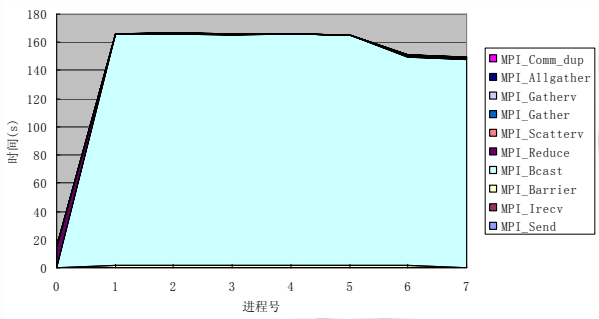



图 6 CCSM3 耦合器的通信子的通信

### 3.5 针对不均衡程序的简单优化

对于不均衡的现象,可以调整 CCSM3 的各个程序的进程数的比例.由于调整时有一定限制,并不是任意比例,因此我们将子程序 4(pop)的进程数调整为 48 个进程(之前为 24 个),子程序 2 (csim)不变,原来计算较少的子程序 1 也就是耦合器(cpl)和子程序 3 (clm)调整为 4 个进程,子程序 5(cam)调整为 20 个进程,得到的计算和通信结果如图 7 和图 8 所示.

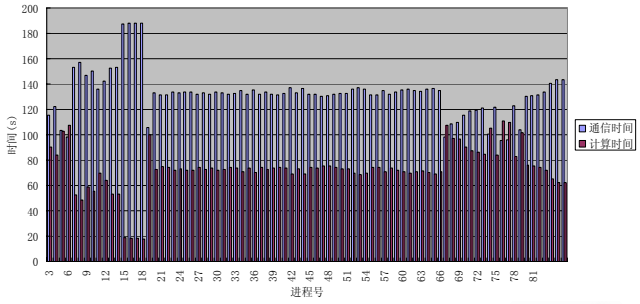



图 7 调整后的 CCSM3 的整体时间统计

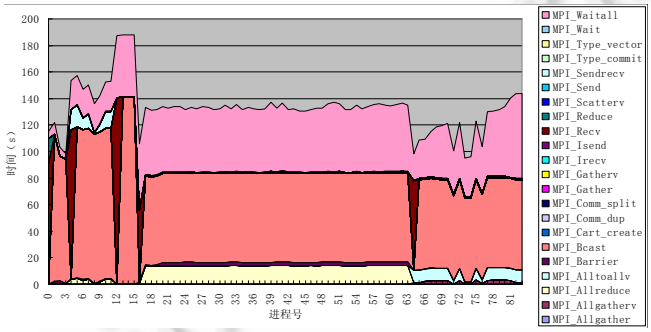



图 8 调整后的 CCSM3 的通信时间统计

调整后的性能提高稳定在 5%左右.不难看出,与调整之前的性能相比,调整后的程序的计算相对之前也要更加均衡,通信上除了子程序因为进程数增加而通信时间增加以外,其余的进程的通信时间都有所减少.这说明我们之前的观察是正确的.虽然从统计结果上看仍然存在较长的 Recv 等待的问题,但是以耦合器为重点来观察程序的整体负载均衡问题是可行的思路,继续优化和调整程序的配置也将是我们未来的工作.

### 3.6 性能采集的开销

性能采集的开销通常决定了结果是否准确有效.我们在采集整体统计和分通信子统计数据时,多次运行性能开销远小于 1%,输出的性能文件不超过 1MB;而在采集耦合器详细的性能时,损失仅为 5/200,即 2.5%左右,而输出文件因为限制于耦合器的采集,仅为 5MB 左右(如果不限制在耦合器的采集,则输出为 370MB 左右),因此,我们所进行的采集并没有对源程序的特征进行干扰,而输出文件大小也得到了很好的优化.

## 4 结 论

MPMD 计算模型是气候计算领域的重要内容,而负载均衡也是并行程序性能分析的重要问题.本文通过分析 CCSM3 程序,阐述了如何更容易地定位 MPMD 程序中特有的子程序间的负载不均衡的现象,并给出了具体的实例,最后通过调整程序配置验证了我们的观点.这表明,通过重点观察耦合器的性能数据,将能更好地确认程序中的负载不均衡作用.这种特性是 SPMD 程序不具有的.利用这种特点,将使性能分析和优化的工作更有方向性,也更加有价值.

**References**:

[1] Top500 List for June 2011. http://www.top500.org/lists/2011/06

[2] 陈国良.并行计算——结构、算法、编程.高等教育出版社.349-351.

[3] Bhandarkar M, Kale L, de Sturler E, Hoeflinger J. Adaptive load balancing for MPI programs. In: Computational Science-ICCS. Springer-Verlag, 2001. 108–117.

[4] Corbalan J, Duran A, Labarta J. Dynamic load balancing of MPI+ OpenMP applications. In: Proc. of the ICPP. IEEE Computer Society, 2004.

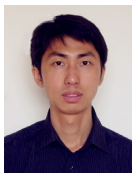
[5] George W. Dynamic load-balancing for data-parallel MPI programs. In: Proc. of Message Passing Interface Developer's and User's Conf. (MPIDC'99). 1999.

[6] Tallent NR, Adhianto L, Mellor-Crummey JM. Scalable identification of load imbalance in parallel executions using call path profiles. In: Proc. of the 2010 ACM/IEEE Conf. on Supercomputing (SC 2010). Washington: IEEE Computer Society, 2010. 1–11.

[7] Collins WD, Bitz CM, Blackmon ML, Bonan GB, Bretherton CS, Carton JA, Chang P, Doney SC, Hack JJ, Henderson TB, Kiehl JT, Large WG, Mckenna DS, Santer BD, Smith RD. The community climate system model Version 3 (CCSM3). Journal of Climate, 2006,19.

[8] Vetter J, McCracken M. Statistical scalability analysis of communication operations in distributed applications. In: Proc. of the ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming. New York: ACM, 2001. 123–132.

[9] Wolf F, Mohr B. Kojak—A tool set for automatic performance analysis of parallel applications. In: Proc. of the European Conf. on Parallel Computing (Euro-Par). Klagenfurt: Springer-Verlag, 2003. 1301–1304.



**吴宏**(1980－),男,江苏扬州人,助理研究员,主要研究领域为并行算法,并行计算.

**翟季冬**(1981－),男,博士,主要研究领域为并行程序的性能分析与预测.

**翟琰**(1987－),男,硕士生,主要研究领域为并行计算,计算机系统评测.