

一种 HPF 程序的监测与分析工具*

都志辉 汪剑平 程旭 许卓群 石利霞

(北京大学计算机科学与技术系 北京 100871)

E-mail: duzh@bigfoot.com

摘要 本文介绍了一个构筑在 HPF(high performance Fortran)编译系统之上的性能监测与性能分析工具.文章概述了 HPF 编译系统,讨论了性能分析的重要性及主要任务,详细介绍了这一工具所使用的性能分析方法和性能监测与性能数据收集方法,并对此工具的使用效果进行了总结.

关键词 性能分析,可视化,HPF,性能监测,并行编译.

中图法分类号 TP314

1 HPF 编译器及其运行支持系统

科学与工程计算中的大部分问题是数据并行问题.HPF(high performance Fortran)^[1,2]是典型的数据并行语言.由于本文所介绍的性能监测与性能分析手段是构筑在我们开发的 HPF 编译系统之上的,故先简单介绍一下这一 HPF 编译系统.

一个编译系统一般分为编译时和运行时两大部分,我们开发的 HPF 编译器将 HPF 程序转换成 FORTRAN77 结点程序,是一种源到源的转换.HPF 编译器包括通用的编译前端和后端变换处理两大部分.前端完成对 HPF 程序的语法分析和静态的语义检查.后端包括程序的规范化、程序分析和程序变换 3 个阶段,完成通信的检测与结点程序代码的产生,对于 HPF 中的每一条可执行语句,都可以在转换后的 FORTRAN77 结点程序中找到对应的程序段,因此很容易对 HPF 程序的每一条并行语句进行监测与分析.运行时使用一个 C++ 开发的类库,它包括分布数组的管理与数据通信两大部分.通过编译时和运行时的接口——分布数组描述子 DAD(distributed array descriptor),接收从编译时传递过来的数组分布信息,完成对分布数组的计算.通信操作是通过在类库中直接调用 MPI^[3]的通信函数来实现的.我们通过精心设计,提出了简单而有效的通信检测算法.我们将通信分为 3 类:无通信、简单移位通信和其他通信,在运行时支持类库中实现了这 3 类通信,并对简单移位通信进行了特殊的优化处理,使它具有很高的通信效率.目前,整个编译系统已能正常运转,对一大类常见问题的 HPF 程序(如 LAPLACE 方程),本编译系统已能产生具有较高执行效率的结点程序,整体运行性能良好.

2 性能分析的主要任务

并行程序设计与串行程序设计有着本质的区别,并行程序设计的编程与调试环境还很不健全,再加上并行编程本身所固有的复杂性,因而并行程序设计的难度很大,编写出来的程序不易调试与维护,因此迫切需要一套协助开发并行程序的工具,即需要并行程序开发环境的支持.高性能是并行程序设计的一个主要目标,因此,并行程序的性能分析工具^[4]就成为并行软件开发环境中一个十分重要的组成部分.

* 本文研究得到国家重大基础研究攀登计划基金和国家 863 高科技项目基金资助.作者都志辉,1970 年生,博士后,主要研究领域为并行计算,科学计算可视化,地理信息系统.汪剑平,1964 年生,博士生,主要研究领域为并行编译,并行算法.程旭,1966 年生,副教授,主要研究领域为并行处理,分布式计算,计算机体系结构.许卓群,1936 年生,教授,博士生导师,主要研究领域为科学可视化,空间数据分析与智能决策,并行计算.石利霞,1973 年生,硕士,主要研究领域为并行编译.

本文通讯联系人:都志辉,北京 100084,清华大学计算机科学与技术系微机教研室

本文 1998-01-20 收到原稿,1998-11-02 收到修改稿

以 HPF 为代表的高级并行语言的出现大大降低了并行程序设计的难度,提高了并行程序的开发效率,但同时也带来了另一个问题,即如何保证高级并行程序的执行效率.程序的性能分析对高级并行程序设计显得尤为重要.具体地说,并行程序性能分析应能回答或帮助程序员回答以下问题.

(1) 该并行程序的性能如何?即对并行程序作总体上的性能评价,从而使程序开发人员知道该并行程序的性能是否达到给定的要求,若未达到要求,则需要进一步回答下面的问题.该程序离要求的性能指标越远,往往越是需要详细的性能信息来帮助程序员提高程序的性能.

(2) 该并行程序在性能方面的主要问题表现在什么地方?即性能定位.给出影响程序性能的主要部分,即找出影响程序性能的瓶颈.对瓶颈部分的改进可以大幅度地提高程序的性能,因此,性能定位是提高性能的重要一环.

(3) 引起该并行程序性能瓶颈的主要原因是什么?即性能的分析^[5].这个问题是第(2)个问题的继续,因为仅仅知道问题在什么地方还不够,还需要知道在该处引起性能问题的具体原因,比如是由于负载不平衡引起的,还是由于通信量过大引起的.原因不同,解决的办法自然也就不同.

以上是并行程序性能分析应该回答的问题,也是它的主要任务.根据我们调试 HPF 程序和 HPF 编译器的经验,性能分析可以从 HPF 源程序和编译得到的 SPMD(single program multiple data)结点程序两个方面来帮助我们提高 HPF 执行程序的性能,同时为改进我们的 HPF 编译器提供定性和定量的指导.

目前,并行程序的性能分析工具可以按其工作方式分为监测型和预测型^[6]两类.监测型工具通过在程序实际运行时收集性能数据,然后进行统计分析.预测型工具要求建立一个数学模型,用若干参数来表示影响程序性能的因素,然后通过程序结构进行静态分析,估计这些参数的值,进而预测程序性能.由于我们的 HPF 编译器是一个正在开发的编译器,许多性能参数还无法确定,因此很难建立一个数学模型来预测程序的性能,故而将我们的性能分析工具设计成监测型的,而不是预测型的.

3 性能分析

性能分析是在性能监测和性能数据收集的基础上进行的,但在叙述上需要先介绍一下我们的性能分析工具的主要方面,然后说明我们是如何针对这些方面对程序的运行进行监测以及如何收集性能数据的.

影响数据并行程序性能的主要因素有以下 3 个方面.

(1) 负载平衡.负载平衡是并行计算特有的也是影响并行程序性能的最主要的因素.所谓负载平衡,即将一个任务划分成几个大体相当的部分,让多个处理器共同承担.若有的处理器负担过重而有的处理器负担过轻,就会失去并行计算的优势,严重影响并行程序的性能.因此与负载平衡有关的信息是搜集并行程序性能信息的一个重要方面.

(2) 通信.并行与通信是一个问题不可分割的两个方面,并行往往引起通信,只有并行而没有通信是极个别的情况.并行可以提高程序的性能,而通信会对程序的性能产生不良的影响,一次通信的开销可能超过几十甚至几百万次计算的开销,因而并行程序设计中必须十分谨慎地处理通信问题,尽可能避免不必要的通信.通信问题又可细分为通信次数过多、通信量过大、通信路径不合理等方面.与通信有关的信息是搜集并行程序性能信息的另一个重要方面.

(3) 计算.为达到一定的计算结果,可以采用不同的计算方法及手段.这往往对程序的性能影响很大,比如将循环体中的一个简单的函数调用改为“inline”直接计算,可以成倍地提高程序的效率.计算也是影响并行程序性能的一个重要方面.

影响数据并行程序性能的因素还有很多.我们的性能分析工具重点对这 3 个方面进行性能监测,根据监测得到的数据进行性能分析.

监测与这些因素有关的信息,并进行可视化(visualization)^[7]是一种重要的分析科学数据的方法.对并行程序性能数据的可视化可以用更直观、更富于语义的方式来展示性能数据.并行程序的性能数据往往数量大,结构复杂,为了有效地帮助程序员来理解和利用这些信息,可视化技术是一种重要的表现手段.

本文介绍的 HPF 可视化性能分析工具是我们整个 HPF 编译系统的一个重要组成部分,它不仅能帮助

HPF 程序开发者改进其 HPF 程序的性能,而且同时为 HPF 编译程序开发者提供帮助,改进编译程序的开发,以产生更有效的 HPF 结点程序。

我们的可视化性能分析工具的系统流程如图 1 所示。它提供如下几个方面的功能:(1) 中间表示结构及其与源程序对应关系的可视化;(2) 数据分布的可视化;(3) 通信方向及通信量的可视化;(4) 计算的可视化;(5) 运行时信息的统计分析。

通过可视化工具,可以展示源程序中任何语句的中间表示结构,如数据分配语句的结构;FORALL 语句的结构;IF,DO 等控制语句的结构等。

所谓数据分布可视化即是对每一分布式数组,将它在不同处理器上的分布情况展示出来,如图 2 所示。数据分布有动态和静态两种情况。有了数据分布信息,便可以粗略地了解各处理器的负载情况,将它和计算结合起来便能更清楚地认识到数据分布的正确性与合理性。数据并行的一个重要方面是数据在不同处理器上的划分,有了数据划分的信息,对于分析提高数据并行性是很有意义的。

关于通信可视化,主要是展示哪两个处理器之间有通信联系,通信量如何,如图 3 所示。通过减少通信或改进通信可以提高效率。这部分的可视化数据是在程序运行后得到的,是程序的动态数据。

关于计算可视化,主要是指各处理器的负载情况,比如将各处理器的负载以 Kiviat 图显示出来,如图 4 所示,有了这些信息,就可以优化计算,提高并行性。

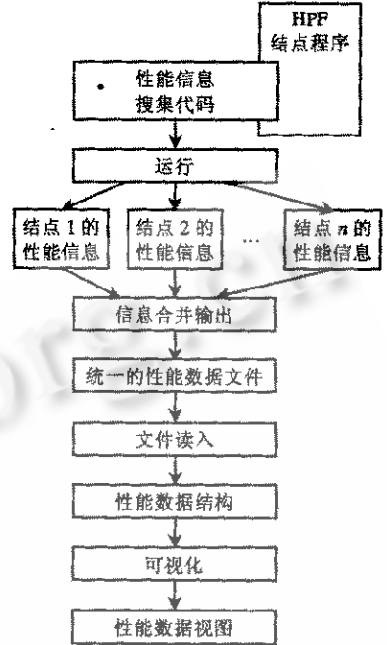


图 1 性能可视化工具系统流程

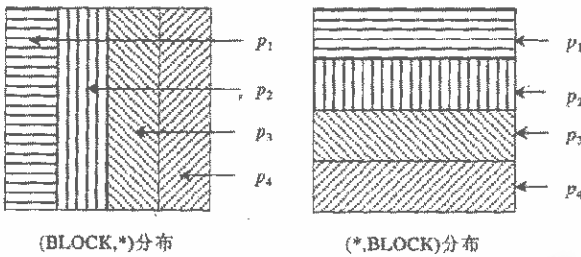


图 2 数据分布示意图

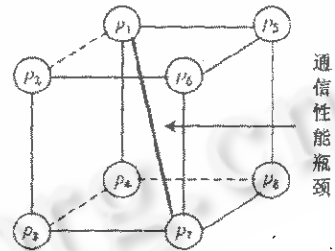


图 3 通信示意图

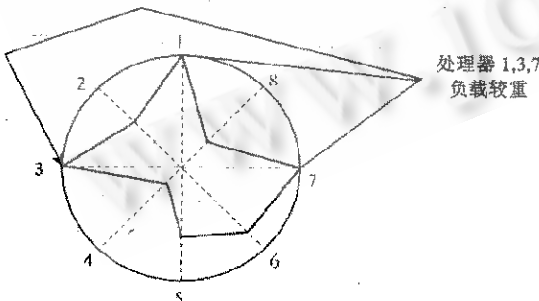


图 4 计算量 Kiviat 图

关于程序的各种统计信息,主要的可视化方法包括表格显示、饼图显示、棒图显示、Kiviat 图显示等,反映各个处理器上执行程序的主要性能指标,包括运行时间和通信状况,为程序员迅速了解程序的总体性能情况提供了一个方便的入口。

通过可视化工具对静态的数据分布及动态的通信和计算进行分析,可以找到性能瓶颈,既为 HPF 程序员,也为 HPF 编译程序开发者提供了改进各自程序的依据,有利于开发出高质量的 HPF 程序及 HPF 编译程序。

4 程序的运行监测及性能数据的收集

具体地说,我们是通过在编译产生的 FORTRAN77 结点程序和运行支持类库中增加性能监测语句来对程序的运行进行监测和性能数据收集的。

A. 监测收集的性能信息分为3类:运行统计信息、运行时序信息和数据分布信息.

① 运行统计信息又分为整个程序的运行统计和各并行语句的运行统计信息.一个并行语句的运行统计信息是对该语句的多次执行的统计结果,包括该语句运行的总时间、总通信量、总循环次数等信息.

② 程序运行时序可以用程序运行过程中状态的转变来描述.我们把每次程序运行状态的转变定义为一个事件,用事件类来描述.每个事件都有一个表示当前程序状态的属性、当前所在语句号以及时间戳.目前定义的程序状态包括串行计算、并行计算、发送、接收和空闲.程序每次在这些状态之间的转换都被记录为一次事件,对发送和接收事件还包括其通信目的、通信量以及通信方式等属性.

③ 在 HPF 中数据分布的定义包括定义虚拟处理器组、模板以及数组的分布和对齐关系.性能分析引入3个类以分别表示虚拟处理器组、模板和数组.对每个处理器都收集这3类信息,结合起来成为一个处理器的全部性能信息,也定义为一个类.

B. 性能信息收集的实现是由一个基于性能信息类库的 C₉₅ 语言函数库 ProfLib 完成的,它提供了一组操纵类库数据结构的函数和一组直接存取的全局变量,可以用 C 语言和 Fortran 语言调用.

① 并行程序运行统计类对象的维护完全在运行支持库中完成,程序运行时间通过 ProfLib 的初始化时间和结束时间相减得到,通信方面的统计信息则由并行语句运行时间加总而得.

② 并行语句运行统计类对象是随程序运行动态生成的,对每一个并行语句创建一次.在 HPF 产生的 FORTRAN77 代码中,进入并行计算部分时调用 ProfPushStack,将当前运行上下文设在该语句相应的对象中,结束该语句时,调用 ProfPopStack,将统计信息保存到该对象中.由于要处理嵌套的情况,所以采用了栈结构.统计信息是由插入到运行支持类库中的监测语句产生的.在 ProfLib 中定义了一组全局变量,对应于并行语句运行统计类对象的各成员,它们类似于一个多任务操作系统中的寄存器组,在 ProfPushStack 时,把现场保存到原栈顶对象中,把新对象的统计信息恢复到寄存器中,以后的统计都在全局变量上直接操作,直到 ProfPopStack 时,再把现场保存至该对象中,恢复原栈顶对象的现场.这样处理的好处在于,性能信息的统计直接操纵全局变量,只在保存和恢复现场时调用类库方法,从而节省了这些调用的开销.

③ 运行事件是运行支持类库中通过调用 ewEvent 函数而动态产生的,ProfLib 的其他函数(例如 ProfPushStack)也可能产生运行事件.在 FORTRAN 77 代码中无需增加语句.

④ 数据分布信息在数据分布定义函数中加以收集,但需要在 FORTRAN 77 代码中插入对 ProfLib 的调用.

性能信息的收集在设计中要遵循的首要原则是尽可能降低收集代码的时间开销,减少对程序性能的影响.因此在设计中采取了一些措施:尽量采用静态数据,减少动态分配;减少函数调用,尽量把必需的函数定义为 inline 型;所有性能信息都在内存中处理,直到程序退出时才一并输出.

5 相关工作

文献[4]提出的是一种监测型的性能分析和调试工具,但它主要针对的是 MPP 平台.我们的工具不仅具有调试功能,而且不限于特定的平台.文献[6]是预测型的性能分析工具,它依赖于对算法和并行机的许多指标准确地量化,通过建立一个简单的性能模型来进行预测.我们计划在现在的监测型工具的基础上,对一些典型的应用平台实现性能预测的功能.文献[8]的工作重点是建立了一个原始代码和优化后代码的对应关系,而性能分析方面的功能较弱,我们的工具的重点放在对性能数据的监测和分析上,而且也实现了原始代码和优化后代码的对应关系.

6 性能分析对编译优化的意义

目前我们使用此 HPF 性能监测与性能分析工具,已对 HPF 编译优化产生了显著作用.通过对 HPF 程序及其结点程序的分析,找到了影响 HPF 程序性能的主要因素.基于此工具得到的结论,我们重新调整了我们的 HPF 编译器的优化重点和优化策略,结点程序的性能有显著的提高,用4台机器测得的不同例子的加速比增加了 0.3~1.2 不等.性能监测与性能分析工具使我们对 HPF 程序及编译得到的结点程序从定性到定量两

个方面都有了比较具体而深刻的认识,而且对编译程序的优化重点也做到了心中有数,这些都归功于这一工具反馈给我们以直观形象的性能信息,使我们能够有针对性地提高 HPF 程序的性能。

致谢 感谢李昱和郑耿斌硕士为本文的工作所付出的辛勤劳动,感谢北京大学计算机科学与技术系并行编译研究组的全体老师和同学。

参考文献

- 1 High Performance Fortran Forum. High performance Fortran language specification. Version 1.0, Houston, Texas: Rice University, 1993
- 2 Fortran 工作组. 标准 Fortran 90 语言程序设计. 北京:学苑出版社,1994
(Fortran Group. Standard Fortran 90 Programming Language. Beijing: Xueyuan Publishing House, 1994)
- 3 Snir M, Otto S, Lederman S H *et al.* MPI: the Complete Reference. New York: MIT Press, 1997. <http://www.mpi-forum.org/>, Message Passing Interface (MPI) Forum Home Page
- 4 Wislser R, Oberhuber M, Krammer S *et al.* Interactive debugging and performance analysis of massively parallel applications. *Parallel Computing*, 1996,22(3):415-442
- 5 Fahringer T. Estimating and optimizing performance for parallel programs. *Computer*, 1995,28(11):47-56
- 6 Driscoll M A, Daasch R W. Accurate prediction of parallel program execution time. *Journal of Parallel and Distributed Computing*, 1995,25(1):16-30
- 7 Heath M T, Malony A D, Rover D T. Parallel performance visualization: from practice to theory. *IEEE Parallel and Distributed Technology*, 1995,3(4):44-60
- 8 Shaun Kaneshiro, Tatsuya Shindo. Profiling optimized code: a profiling system for an HPF compiler. In: Proceedings of the 10th International Parallel Processing Symposium. 1996. 469-473

A Tool for Monitoring and Profiling HPF Programs

DU Zhi-hui WANG Jian-ping CHENG Xu XU Zhuo-qun SHI Li-xia

(Department of Computer Science and Technology Beijing University Beijing 100871)

Abstract In this paper, the authors introduce a monitoring and profiling tool which is used in a high performance Fortran (HPF) compilation system. A brief overview of the HPF compilation system is given first, then the importance and main tasks of performance analysis are discussed, the method of performance profiling and monitoring and the method of performance data collecting are given in detail. The paper is concluded with a description of the tool's role in the HPF compilation system.

Key words Performance analysis, visualization, high performance Fortran (HPF), performance monitoring, parallel compilation.