

并行计算性能的分析 and 预测^{*}

Parallel Performance Analysis and Prediction

孟杰 王小鸽 李三立

(清华大学计算机系 北京 100084)

Abstract The incremental use of cluster system in scientific computing has generated a need for tools that can manage the environment, monitor the execution of parallel application, analyze and predict its performance when the problem size and number of nodes changed. We have designed and developed a performance system, THPT, to meet these requirement. In this paper, we have discussed the performance analyzing and predicting subsystem: PAPS.

Keywords Performance analysis, Performance prediction, Cluster system

基于分布存储的工作站机群系统由于具有很高的性能价格比,越来越受到高性能计算领域的重视。用户在使用机群系统时,迫切地需要一种性能监视、评估和预测的工具。THPT 系统就是为满足这一需要而设计开发的。

THPT 系统由两个部分组成:网络并行计算环境的实时管理和性能监测子系统 MMS 以及并行计算性能分析和预测子系统 PAPS。本文主要对 PAPS 子系统进行论述。

1. 机群系统和 MPI 消息传递平台

机群系统是由多个独立的结点计算机通过高速网络互联得到的。在运行并行应用时,多个结点机之间通过消息传递机制进行协调。在结点机的操作系统之上有一个消息传递平台。常用的消息传递平台有 MPI 和 PVM。

我们的机群系统由若干高性能的 Pentium 机组成,它们通过高速局域网互连在一起。每个结点机上运行 LINUX 操作系统,并采用 MPI 的 MPICH 版本作为消息传递的软件平台。

在 1993 年,并行计算公司、软件编写者与应用科学家合作编写了一个标准的可移植的消息传递库,称为 MPI(Message Passing Interface)。MPI 是

一个例行程序库的规范。在 MPI 中,提供了一系列消息传递函数,以支持并行计算环境中参加并行计算的程序之间的通信。

随着应用的需要和发展,在 MPI 标准的基础上,加入了一些额外的例行程序库,称为 MPE(MultiProcessing Environment),它最终成为扩展后的 MPI 标准的一部分。MPE 包括了一套产生运行记录文件 logfile 的例行程序,这些 logfile 可用来做性能分析和性能可视化。Argonne 国家实验室开发的性能分析工具 Nupshot 就是根据 logfile 中提供的信息完成其分析功能的。

2. 性能分析和预测子系统(PAPS)

THPT 系统有两个组成部分,其中 MMS 是一个实时系统,主要为用户提供以图形方式管理、使用机群系统,并在并行应用运行过程中对其进行实时监测。而 PAPS 是一个运行后系统,它根据并行应用运行产生的记录文件(logfile)进行性能分析和性能预测。

在机群系统中,各个组成结点机具有各自独立的系统配置,它们除了参加并行计算外,还可以同时进行各自的工作。另外,并行应用本身分配给各个参

^{*} 本文受东南大学开放实验室基金和国家攀登计划资助。孟杰 硕士,讲师,主要研究方向是网络并行计算、分布式计算环境。王小鸽 博士,副教授,主要研究方向是并行算法。李三立 教授,博士生导师,主要从事新型 RISC 体系结构、指令级并行处理和并行处理系统等方面的研究工作。

加结点机的工作负载也可能不同,这样,当并行应用的性能不理想时,如何找到运行瓶颈,以及如何在特定的并行环境中更好地改进并行算法,以达到最佳的并行性能就显得尤其重要。

并行应用具有运算量大的特点。在并行应用开发过程中,用户通常使用少量结点机进行调试,调试时也只能使用问题规模较小的例子,这时,用户便希望能够拥有一个能够根据小规模实验预测大规模应用的性能的工具。这里的规模包括问题规模的大小以及参加并行计算的结点机的个数。

PAPS 子系统就是为了满足用户的上述需求设计的,它提供了对并行应用进行性能分析,并根据小规模应用的实验数据预测大规模应用的性能的功能。它主要是根据应用运行时产生的 logfile 来进行性能分析和预测的,其中的性能分析工具 TH-Nupshot 根据 logfile 生成形象的画面,使得用户能够对并行应用运行过程中所经历的各种运算状态(包括计算、通信、同步等),各状态所持续的时间以及相应各个时间段中系统资源利用率有一个整体的了解,以分析瓶颈发生的位置和原因。性能预测部分则根据 logfile 抽象出反映应用结构的应用逻辑树(ALT),并据此对更大规模的并行应用进行性能预测。THPT 系统中 PAPS 子系统的结构如图 1 所示。

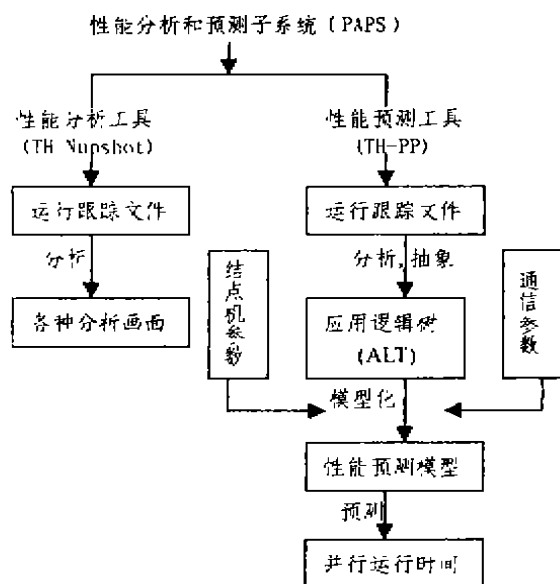


图 1 性能分析和预测子系统的结构设计

3. 并行性能的分析

性能分析工具可以帮助用户了解并行应用的运

行过程,找出影响性能的瓶颈,使得并行算法和并行计算环境得到更好的匹配,从而得到最佳的并行性能。

并行计算性能主要受以下因素的影响:并行环境的通信性能、各个结点机的计算性能、并行应用所采用的并行算法。由于机群系统中各个结点机的资源配置可能不同,工作负载也不尽相同,所以它们的系统资源利用情况也是影响性能的主要因素。作为并行性能分析工具,应该为用户形象地提供这些信息,包括并行应用运行过程中消耗在通信、同步、计算等过程的时间,以及在這些过程中,结点机上的系统负载情况等。

Argonne 国家实验室提供了性能分析工具 Nupshot,它根据并行应用运行时产生的 logfile,为用户提供一个可视化的分析。PAPS 系统中的性能分析工具 TH-Nupshot 主要是在 Nupshot 的基础上加以适当扩充得到的。首先,它扩充了并行应用运行中产生的 logfile 文件中所包含的信息,将各个运算时期并行应用在各个结点机所占用的系统资源信息添加进去,在产生可视化画面时,就可以为用户提供系统负载资源利用方面的信息。

在 TH-Nupshot 提供的可视化画面中,一个结点机对应两条时间轴,第一条时间轴上主要显示该结点机在并行应用执行过程中的各运行状态及其持续的时间。不同的状态用不同的颜色来表示,而各个颜色块的长度则代表持续时间。另一条时间轴描述该结点机在应用运行时的系统资源利用信息,包括 CPU 和内存的利用率。利用率的多少用长方条的高度来表示。用户参照时间轴可以一目了然地看到在某一时刻结点机所处的运行状态。用户可以通过点击鼠标来查看所关心的某个状态的详细信息。

4. 性能预测模型

我们使用一种较概括的模型来预测并行计算性能,从而使用户能够根据小规模应用的性能来预测大规模应用的性能。在机群系统中,并行计算性能是由并行应用中的计算和通信延迟以及并行算法决定的。计算延迟由计算量、进行计算的结点机的性能参数以及结点机的资源利用情况决定。而通信延迟则由结点机之间的通信量和通信性能决定,并行应用通过一定的算法将计算量分配到各个结点机中,并通过消息传递函数在各个结点机之间传递数据,以保证并行计算的顺利进行。因此,进行性能预测需要对结点机性能、通信性能以及由并行算法决定的计

算与通信的前后关系进行模型化。

4.1 结点计算性能模型

单个结点机的计算性能是由结点机的 CPU 频率、内存容量以及结点机上的负载情况决定的,我们用一种抽象的并行机器模型的概念来将结点机的计算性能模型化。这种抽象的并行机器能够直接执行较高级的语言指令(HLLI)。可以用结点机执行 HLLI 的时间来描述结点机的计算性能。

在并行应用中,计算部分都是由一个个基本计算单位(BCU,如一个迭代步、一次向量乘等)组成的。我们把这种 BCU 当作更高级的 HLLI。结点机的计算性能就可以用结点机对各种经常用到的 BCU 的执行时间组成的向量 V_m 来描述。

4.2 结点机间的通信性能模型

在网络并行计算系统中,结点机之间的通信性能是影响网络并行计算性能的主要因素,因此,对结点机之间的通信性能进行模型化在并行计算性能的预测中起着非常重要的作用。点对点通信中,消息传递的通信延迟是由消息长度 m 、网络带宽 B 和软件初始开销 T_0 决定的。可以表示为: $T_{comm} = T_0 + m/B$ 。对集合通信和集合计算而言,通信时间还受到参加并行计算的结点机数的影响,其通信时间可以表示为: $T_{comm} = T_0(P) + m/B * f(P)$ 。

4.3 并行应用的逻辑树模型

并行应用的各个计算部分是按照一定的算法分配到各结点机中的。分配到各个结点机中的各计算部分之间的先后关系及通信关系是影响并行计算性能的主要因素。我们可以将并行应用中的这种先后关系和通信关系用一个有向树表示出来,称为应用逻辑树(ALT),ALT 可以由并行应用的预编译器产生,也可以从并行应用的小规模运行产生的跟踪文件中抽象出来。

在 ALT 中,结点代表各结点机的计算状态,用基本计算单位的数量列表 C_m 来表示计算量,这时,每一个结点对应的计算延迟为 $T_{comp} = C_m * V_m * F_s$ (其中, F_s 表示系统资源利用率对计算时间的影响),ALT 中的边代表结点机之间的通信,用传递的消息长度来表示通信量,对应的通信延迟为 $T_{comm} = T_0(P) + m/B * f(P)$ 。这样,ALT 便成为一个有向加权树。在 ALT 中,必有一条路径花费的时间最长,我们称这条路径为关键路径。该主要路径的时间即为并行应用的执行时间。这样,在 ALT 中查找主要路径,并计算其消耗的时间,就可以预测出并行应用的执行时间。

• 16 •

5. 性能预测工具 TH-PP

并行应用的运行记录文件 logfile 中,记录了并行应用所经历的各种状态的信息以及各个通信状态之间的消息传递关系。根据这些信息,我们可以从中抽象出并行应用的抽象逻辑树(ALT),并根据 ALT 进行性能预测。

使用 TH-PP 进行预测时,根据 logfile 中生成的 ALT 结构保存在数据结构中。我们用嵌入 C 的 Tcl/Tk 语言实现 ALT 的显示,以便为用户提供一个直观的画面。在画面中结点表示每个结点机的计算状态,每条带箭头的线段表示各结点机间数据发送、接收的过程。用户可以通过点击鼠标来查看各点和边所对应的详细信息,如点所对应的计算时间、边所对应的通信信息的长度等。

当在处理机结点数不变的情况下,用小问题规模的 ALT 来预测问题规模扩大时的并行性能时,预测可以分成三个部分:对 ALT 中计算结点的计算时间的预测;对 ALT 中边的通信时间的预测;ALT 中最长路径的查找。

我们可以通过实验数据得出在系统空闲时常用到的基本运算单位 BCU (如矩阵乘、向量乘等)的执行时间。这些数据可以代表各个结点机的计算性能。当问题规模扩大时,可以根据所用到的各种 BCU 的数目的变化、系统资源的利用情况(如 CPU 利用率、内存容量等)来预测出问题规模扩大后 ALT 图中各个计算结点所对应的计算时间。ALT 图中边所对应的通信时间可以通过通信模型以及改变后的消息长度来进行预测。

结点代表的计算时间预测和边代表的通信时间预测结束后,ALT 作为加权有向树,权重已经有所改变。重新计算 ALT 有向树中的最长路径,便可以预测出并行计算时间。

当结点机个数增加时,为简化讨论,我们只考虑 SPMD 的并行程序,这时,总的计算量被分配到更多的结点中去,使得单个结点机承担的计算量减少,结点机之间的通信量也相应减少。但由于各个结点机的配置可能不同,对应于 ALT 图中各个结点的计算时间也可能不同。这样,可以产生一个新的 ALT 结构图。通过查找这个 ALT 图中的最长路径,便可以预测出并行计算时间。

性能预测实例 泊松问题是较普遍的偏微分方程问题。我们以它为例,对并行计算的性能进行预测。泊松问题的方程表示为:

$$\partial^2 f(x, y) / \partial x^2 + \partial^2 f(x, y) / \partial y^2 = 0$$

我们定义一个矩形网络(grid)来求偏微分方程的近似解。其 Jacobi 迭代的离散形式为:

$$f_{\text{new}}(i, j) = (f(i-1, j) + f(i+1, j) + f(i, j-1) + f(i, j+1)) / 4$$

计算函数点的新值时用周围函数点的旧值。并行程序在给定边界的情况下,对其进行 Jacobi 迭代,然后判定其是否收敛。

在对泊松问题进行并行计算时,我们对该矩阵进行一维划分。每个结点机分得一个子阵。对于阵的边缘结点进行迭代时,需要与相邻的结点机进行通信。

在结点机数目为 4 的情况下,实际的计算时间与使用性能预测工具得到的预测时间的比较如图 2 所示。

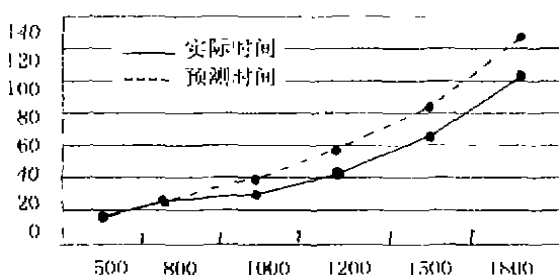


图 2

我们发现随着规模的增加,预测时间与实际执行时间的偏差增大,这是因为误差的累积。在实际运算中,并行计算的速度还受到系统进程、内存容量等其他因素的影响。另外一个因素是同步函数 Barrier 和 AllReduce 等集合操作的影响,这些状态在预测

模型中没有考虑,数据规模越大,这些因素的影响越大。

结束语 目前并行性能分析和预测工具主体已经完成。并行性能预测工具 TH-PP 尚需进一步完善。问题规模的扩大和系统资源利用率的关系、可伸缩性分析是将来需重点探讨的两个问题。

参考文献

- 1 Miller B P, et al. *The Paradyn Parallel Performance Measurement Tools*. IEEE Computer, Special issue on performance evaluation tools for parallel and distributed computer systems, 1995, 28(11)
- 2 Martonosi M, et al. *Tuning Memory Performance of Sequential and Parallel Programs*. IEEE Computer, 1995(4): 32~39
- 3 Calzarossa M, et al. *Medea: A Tool for Workload Characterization of Parallel Systems*. IEEE Parallel & Distributed Technology, 1995 Winter: 72~83
- 4 Boden N J, et al. *Myrinet: A Gigabit per Second Local Area Network*. IEEE Micro, 1995, 15(1): 29~36
- 5 Crovella M E, LeBlane T J. *The Search for Lost Cycles, A New Approach to Parallel Program Performance Evaluation*. [Technical Report 477] The University of Rochester, Computer Science Department, Rochester, New York 14627, 1993
- 6 Sarukkai S R, Mehra P. *Automated Scalability Analysis of Message-Passing Parallel Programs*. Same to [3]: 21~32
- 7 Borgeest R, et al. *A Trace Based Performance Evaluation Tool for Parallel Real Time Systems*. Parallel Computing, 1995, 21: 551~564
- 8 Clifford J. *A Model for Historical Databases*. In: Proc. of Workshop On Logical Bases for Database. Toulouse, France, 1982
- 9 Gadia S K. Temporal element as a primitive for time in temporal databases and its application in query optimization. In: Proc. of 1986. ACM Computer Science Conf.
- 10 唐常杰, 熊明. The Temporal Mechanisms in HBase. J. of Computer Science and Technology, 1996, 11(4): 365~371
- 11 唐常杰, 于中华, 等. 时态数据库的分史存储技术. 第 15 届全国数据库会议论文集, 1998

(上接第 29 页)

- 3 Ginsburg S, Tang Changjie (唐常杰). Projection of Object History. Theoretical Computer Science, 1986, 48: 297~325
- 4 Ginsburg S, Tang Changjie (唐常杰).
- 5 Novathe S B, Ahmed R. A Temporal Model and Query Languages. [UF-CIS Technical Report TR-8516-Computer and Information Science]. Department Univ. of Florida, April 1986
- 6 Snodgrass R, Ahn I. Temporal Databases. IEEE Computer, 1986, 19(9): 35~42
- 7 Tansel A U. Adding time dimension to relational model and extending relational algebra. Information Systems, 1986, 11(4): 343~353