

基于工作站集群环境的可扩放性度量标准

A Scalability Metric Based on NOW

丁卫群 计永超 陈国良

(国家高性能计算中心 中国科学技术大学计算机系 合肥 230037)

Abstract Along with the rapid development of parallel processing technology and the popularity of NOW, the scalability of parallel algorithm-machine combinations, which measures the capacity of a parallel algorithm to effectively utilize an increasing number of processors, becomes more and more important. In this paper, we present a new metric, called time-scale metric, to measure and evaluate the scalability of parallel algorithms and machines, and extend it to fit the characteristics of NOW. The experimental results show that the time-scale metric is a practical and accurate method to evaluate the scalability of parallel algorithms on MPP and NOW.

Keywords Scalability, Network of workstation, Dawning-1000 MPP, MPI

1. 引言

可扩放性是指并行算法有效利用可扩充的处理机数目的能力,目前已经提出了许多可扩放性度量方法,其中最典型的是:等效率方法^[1]、等平均速度方法^[2]和平均延迟方法^[3]。等效率的方法严格地说只是一种分析的方法,在实际应用中不够准确,而且该方法给出的是工作量与处理器数的关系函数,反映了工作量随处理器数变化的趋势,并没有一个量化的数据。等平均速度的方法将平均速度作为衡量可扩放性的主要指标,是一种将算法与机器相结合的基于测量的方法,但是在实际情况中很难精确地测量出程序运行的速度。平均延迟的方法使用平均计算延迟作为衡量可扩放性的主要指标,精确地考虑了算法与体系结构两者的特性,也是一种基于测量的方法,但该方法需要使用专用的硬件或者专门的系统级软件来测量并行程序运行时每个处理器上的延迟时间,因此难以广泛地应用于各种并行机上。

另外,随着机器的速度和网络技术的飞速发展,工作站集群(NOW)这种新出现的并行计算环境越来越广泛地成为并行计算的平台。但是由于NOW存在着节点的异质性和非独占性等特性,而以前的可扩放性度量方法都没有考虑到这些特性,因此,我们需要一个基于工作站集群的既便于测量又比较精确的可扩放性

度量方法。

在本文中提出了时间比例的可扩放性度量方法。在第二节给出了该方法的定义;在第三节讨论了该可扩放性度量方法与其它方法的关系;在第四节给出了在曙光1000和NHPCC工作站集群上的实验数据。

2. 定义

对于运行于某一给定体系结构机器上的给定算法,用 $T(W, P)$ 表示在 P 个处理器上解决工作量为 W 的问题所需的并行执行时间,用 $T(W', P')$ 表示在 P' ($>P$)个处理器上解决工作量为 W' 的问题所需的并行执行时间。当处理器数目由 P 变化到 P' ,而并行执行的效率 E ($0 \leq E \leq 1$)保持不变,则定义时间比例可扩放性度量为:

$$\text{time-scale}(E, (P, P')) = \frac{T(W, P)}{T(W', P')} \quad (1)$$

在实际情况下,该公式的值小于或者等于1。若值很大(接近于1),就标志着该算法与机器的组合具有良好的可扩放性。相反,若值很小,就标志着需要增加很大的工作量以维持并行效率不变,即可扩放性不好。

由于NOW可以是一个异构型的计算机集群系统,结点的性能各不相同,它存在着节点的异质性和非独占性等特性,因此在NOW上应该重新定义加速比和效率。我们考虑的是整个计算机集群系统的加速比和效率,而不是针对系统中某一个结点的。

丁卫群 硕士,讲师,研究方向:并行与分布式计算,软件工程和分布式数据库。计永超 博士,讲师,研究方向:高性能计算,并行与分布式算法、并行环境和工具等。陈国良 教授、博导,研究方向:并行/分布式/网络计算,并行算法/体系结构和程序设计等。

假设一个计算机集群系统中有 p 个结点, 一个最优的串行算法(程序)在各个结点上单独串行执行所需的最短时间分别记为 $t_i (1 \leq i \leq p)$, 而该算法的并行程序在 p 个结点的 NOW 上运行的时间记为 T_p , 则可定义该系统的效率如下:

$$E = \frac{\sum_{i=1}^p t_i / p}{p * T_p} \quad (2)$$

则利用公式 1 和公式 2, 可以得到在工作站集群环境下的可扩放性度量方法。

3. 时间比例方法与其它方法的关系

可以看出, 时间比例度量方法最大的好处是便于测量, 因为它是使用程序执行的时间作为衡量可扩放性的主要指标, 可以很容易地测量出来, 是一种简单可行的方法, 并且能比较准确地反映出算法-机器组合的可扩放性, 因而能广泛地应用于各种体系结构的并行机, 尤其是能很好地应用于工作站集群系统 NOW 这样一个新兴而又特殊的并行计算环境。

定理 1 时间比例可扩放性度量方法在理论上与 Iso-efficiency、Iso-speed 和 Latency 这三种典型的可扩放性度量方法是一致的。

证明: (1) Iso-efficiency 方法是在效率 E 保持不变的情况下, 工作量 W 随处理器数 N 增加而变化的函数; 时间比例方法是要求保持效率 E 不变的情况下的并行时间的比值; 所以它们的理论基础是一致的, 只不过它们选取的角度不同。

(2) 在 Iso-speed 方法中, 设运行在 P 个处理器上的工作量为 W 的程序串行执行时间为 $W * t_c$, 则效率

$$E = \frac{W * t_c}{P * T(W, P)}, \text{ 因为平均速度 Speed} =$$

$$\frac{W}{P * T(W, P)} \text{ 保持不变, 即: } \frac{W * t_c}{P * T(W, P)} =$$

$$\frac{W' * t_c}{P' * T(W', P')} = \frac{T(W, P)}{T(W', P')} = \frac{P' * W}{P * W'}, \text{ 从而得到时间比例度量公式。}$$

(3) 由时间比例公式可得:

$$\begin{aligned} \frac{T(W, P)}{T(W', P')} &= \frac{T(W, P) - T(W, P) * E}{T(W', P') - T(W', P') * E} \\ &= \frac{T(W, P) - \frac{W * t_c}{P}}{T(W', P') - \frac{W' * t_c}{P'}} = \frac{L(W, P)}{L(W', P')} \end{aligned}$$

从而得到 Latency 度量方法的公式。

4. 时间比例可扩放性度量的实例

从我们所述的时间比例可扩放性定义以及可扩放性度量公式(公式 1)可以看出, 关键是要得出并行程序的执行时间和执行效率之间的关系, 这里借助图

方法来得到两者之间的关系。如图 1 所示, 每一条曲线代表对于一个固定的处理器数 P , 随着工作量 W 的增大, 程序执行的时间和效率的关系曲线。对于给定的 E' , 在图中作一条平行于横坐标的直线, 直线与各条曲线的交叉点所对应的横坐标的值即为并程序的不同的处理器数目时的运行时间, 且效率 E' 保持不变。这样就可以得出在执行效率维持不变的情况下, 对应于不同处理器数目的执行时间, 从而根据可扩放性定义得出处理器数从 P 增加到 P' 的可扩放性(保持效率 E' 不变)。

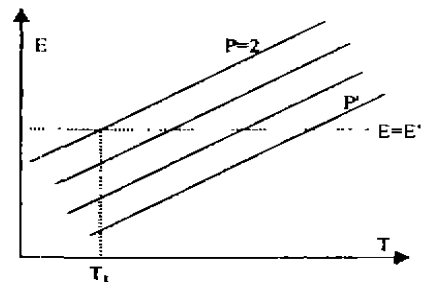


图 1 时间比例可扩放性度量示意图

4.1 在曙光-1000MPP 平台上的实例

国产曙光-1000^[4]是一种分布式存储的大规模并行计算机系统, 它拥有 32 个计算节点, 我们在它上面实现了 MP^[5]环境。

4.1.1 矩阵分块相乘算法^[6] 表 1 给出了当处理器数分别为 1, 4, 9, 16 和 25 时矩阵乘法在曙光 1000 上运行的时间(效率 $E' = 0.76$), 再按照时间比例可扩放性公式(公式 1)可以得到该算法-机器组合的可扩放性, 如表 2 和图 2(当 $P' \geq 4$ 时 $P=4$ 与 $P=1$ 的曲线是重合的)。可见该矩阵乘法在曙光 1000 并行机上的可扩放性是比较好的。

表 1 曙光 1000 上矩阵分块相乘算法运行时间
($E' = 0.76$)

处理器数目	1	4	9	16	25
执行时间(s)	0.84	0.84	4.82	8.15	12.23

表 2 曙光 1000 上矩阵分块相乘算法
可扩放性表示($E' = 0.76$)

Time scale ($E, (P, P')$)	1	4	9	16	25
1	1.0000	1.0000	0.1743	0.1031	0.0687
4		1.0000	0.1743	0.1031	0.0687
9			1.0000	0.5914	0.3941
16				1.0000	0.6664
25					1.0000

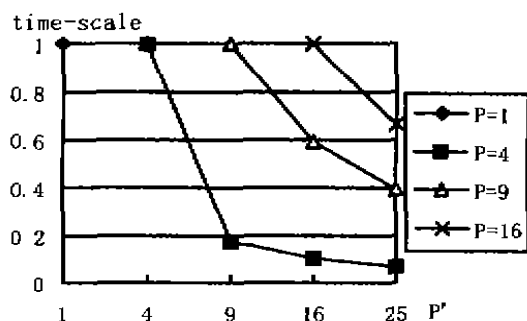


图2 曙光1000上的矩阵分块相乘算法可扩放性曲线图($E'=0.76$)

4.1.2 PSRS 排序算法^[7] 表3给出了当处理器数分别为1,2,4,8,16和32时PSRS算法在曙光1000上运行的时间(效率 $E'=0.65$),再按照时间比例可扩放性公式(公式1)可以得到该算法-机器组合的可扩放性,如表4和图3(当 $P'\geq 2$ 时 $P=2$ 与 $P=1$ 的曲线是重合的)。可见PSRS在曙光1000并行机上的可扩放性一般。

表3 曙光1000上PSRS算法运行时间($E'=0.65$)

处理器数	1	2	4	8	16	32
执行时间(s)	0.1	0.1	0.17	0.53	1.80	2.34

表4 曙光1000上PSRS算法的可扩放性表示($E'=0.65$)

Time scale ($E, (P, P')$)	1	2	4	8	16	32
1	1.0000	1.0000	0.5882	0.1887	0.0556	0.0427
2		1.0000	0.5882	0.1887	0.0556	0.0427
4			1.0000	0.3208	0.0944	0.0726
8				1.0000	0.2944	0.2265
16					1.0000	0.7692
32						1.0000

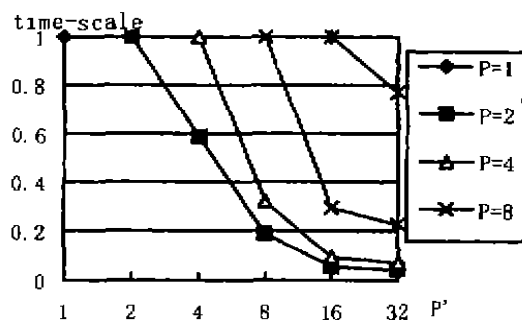


图3 曙光1000上PSRS算法可扩放性曲线图($E'=0.65$)

4.2 在NHPCC-NOW平台上的实例

NHPCC-NOW^[8]是一个工作站集群,它由16台Sun Sparc 20工作站(6台主频为75MHz,4台主频为60MHz,6台主频为50MHz,内存32M),利用100M的SWITCH组成。我们在NHPCC工作站集群环境上也移植了MPI环境。

4.2.1 矩阵分块相乘算法 表5给出了当处理器数分别为1,4,9和16时矩阵乘法在NHPCC-NOW上运行的时间(效率 $E'=0.61$),按照时间比例可扩放性公式(公式1)可以得到该算法-机器组合的可扩放性,如表6和图4(当 $P'\geq 4$ 时 $P=4$ 与 $P=1$ 的曲线是重合的)。可见该矩阵乘法在NHPCC-NOW上的可扩放性不好。

表5 NHPCC-NOW上矩阵分块相乘算法运行时间($E'=0.61$)

处理器数	1	4	9	16
执行时间(s)	1.59	1.59	9.29	32.33

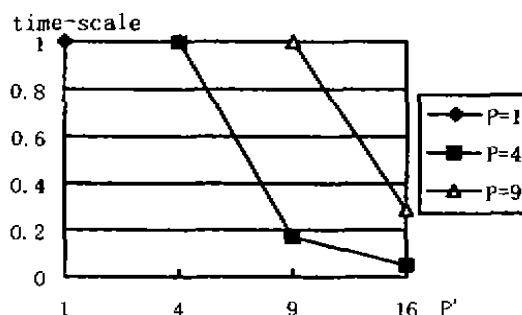


图4 NHPCC-NOW上矩阵分块相乘算法可扩放性曲线图($E'=0.61$)

表6 NHPCC-NOW上矩阵分块相乘算法可扩放性表示($E'=0.61$)

Time scale ($E, (P, P')$)	1	4	9	16
1	1.0000	1.0000	0.1716	0.0492
4		1.0000	0.1716	0.0492
9			1.0000	0.2873
16				1.0000

4.2.2 PSRS 排序算法 表7给出了当处理器数分别为1,2,4,8和16时PSRS算法在NHPCC-NOW上运行的时间(效率 $E'=0.43$),再按照时间比例可扩放性公式(公式1)可以得到该算法-机器组合的可扩放性,如表8和图5(当 $P'\geq 2$ 时 $P=2$ 与 $P=1$ 的曲线是重合的)。可见PSRS在NHPCC-NOW上的可扩放性不好。

表7 NHPCC-NOW 上 PSRS
算法运行时间($E'=0.43$)

处理器数	1	2	4	8	16
执行时间(s)	1.89	1.89	46.67	521.99	880.41

表8 NHPCC-NOW 上 PSRS 算
法可扩展性表示($E'=0.43$)

Time scale ($E, (P, P')$)	1	2	4	8	16
1	1.0000	1.0000	0.0405	0.0036	0.0036
2		1.0000	0.0405	0.0036	0.0021
4			1.0000	0.0894	0.0530
8				1.0000	0.5929
16					1.0000

参考文献

- 1 Kumar V, Gupta A. Analyzing Scalability of Parallel Algorithms and Architectures In: Proc. of the 1991 Int Conf. on SuperComputing, Cologne, Germany, June 1991
- 2 Zhang X D, Yan Y, He K. Latency Metric: An Experimental Method for Measuring and Evaluating Parallel Program and Architecture Scalability. Journal of Parallel and Distributed Computing, 1994, 22: 392~410

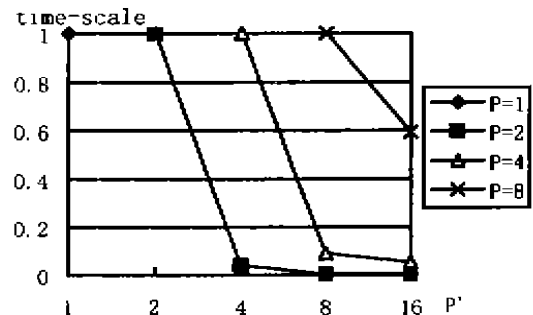


图5 NHPCC-NOW 上 PSRS 算
法可扩展性曲线图($E'=0.43$)

- 3 Sun X H, Rover D T. Scalability of Parallel Algorithm-Machine Combinations. IEEE Trans. Parallel Distributed Systems, 1994, 5(6): 599~613
- 4 曙光-1000 软件系统技术报告. 国家智能计算机研究开发中心, 1995
- 5 Message Passing Interface Forum. MPI: A message-passing interface standard. International Journal of Supercomputer Applications, 1994, 8(3/4)
- 6 陈国良. 并行算法设计与分析. 北京: 高等教育出版社, 1994
- 7 Shi H, Schaeffer J. Parallel Sorting by Regular Sampling. Parallel and Distributed Computing, 1992, 14(4): 316~372
- 8 Ji Yongchang Cluster of NHPCC: [Technique report] CS of USTC, 1997

(上接第 11 页)

分子计算机的基本计算方法是把一种物理识别作为运算的技术, 其特点如下:

(1) 分子计算由蛋白质和它们所综合的系统结构实现, 即不需要象数字电子计算机那样编程;

(2) 分子活动的大规模并行特征可以提高计算效率, 故比电子开关慢五个数量级的单分子速度不会影响整体的运算效率;

(3) 分子计算机属于一种基质感触识别, 因此它依照几何的方法可以实现与上下文有关的自适应信息输入, 而无需按位串行处理。

分子计算机的实现途径可分为两种:

(1) 分子计算机受目前电子计算机的启发, 从分子到电子线路设计与制作出发, 形成一种体积小、速度快、存储量大而成本低的数字分子计算机;

(2) 分子计算机模仿生物的图像信息处理功能, 即快速的模式识别、自组织和自学习能力。

光神经计算机和分子计算机的研究程度及其成果是无法和电子计算机相比拟的, 然而它所发现的新颖计算理论与信息处理的结构体系已引起了生物学家、化学家、物理学家以及计算机学家的极大兴趣, 这会对智能计算机的研究具有极其深远的意义。

参考文献

- 1 Nielsen R H. Neurocomputing; Picking the Human Brain. IEEE Spectrum, Mar. 1988: 36~41
- 2 Perkov N. Systolic simulation of multilayer, feed forward neural networks. Parallel Processing in Neural Systems and Computers. In: R. Eckmiller, G. Hartmann and G. Hauske, eds. Elsevier Science Publishers B. V (North-Holland) 1990: 303~306
- 3 Duncan R. A survey of parallel computer architectures. IEEE computers, 1990, 23(Feb.): 5~16
- 4 Hecht-Niesen R. Performance limits of optical, electrooptical, and electronic neurocomputers. SPIE optical and hybrid computing, 1986, 634
- 5 Kuczewsk R, Myers M, Crawford W. Neurocomputer workstation and processors: Approaches and Applications. IEEE Intl. Conf. Neural Networks, 1988, 3: 487~500
- 6 Hecht-Niesen R. Neurocomputer Interfaces and Performance Measures. IEEE Symp. Circuit and System, 1989: 74~77
- 7 Hecht-Niesen R. Neurocomputing. Addison-Wesley Publishing Company, Inc., 1990
- 8 庄镇全, 王煦法, 王东生. 神经网络与神经计算机. 电子科学技术, 1990(4~11)
- 9 庄镇全, 王煦法, 王东生. 神经网络与神经计算机. 科学出版社, 1992
- 10 戴葵. 神经网络实现技术. 国防科学大学出版社, 1998
- 11 王汝笠, 章明, 周斌. 第六代计算机——人工神经网络计算机. 科学技术出版社, 1992
- 12 Simon Y F, Lisa R A, Yoshiyasu T. Analog Components for the VLSI of Neural Networks. IEEE Circuits and Devices, July 1990