数据并行的性能分析*

刘德才 王鼎兴 沈美明 郑纬民

(清华大学计算机系,北京 100084)

摘要 计算机是一种工具.作为工具,其应用的成功与否具有重要的意义.本文从应用的角度分析了数据并行处理方式下并行处理的性能.首先,本文建立了一个性能分析模型,之后,基于此模型,对影响并行处理性能的各因素进行了详细地分析.本文的分析结果对于并行算法的设计者和并行计算机系统的设计者均具有指导意义.

关键词 并行处理,性能分析,加速比.

由于单机处理速度在可预见的将来不会有很大的提高,因此,对于那些计算复杂性很高的问题(Grand Challenge Problems),甚至一些大计算量的时实问题(如中期数值天气预报),其求解的唯一途径就是多机并行处理.随着微电子和基础计算机科学技术的发展,以及人们对并行处理技术的研究,使得并行处理逐渐走向实用,并成为当前,也将成为今后计算机领域的研究热点之一.目前,并行处理吸引了世界上很多著名的机构和著名的计算机工作者.世界各国在并行处理上均投资巨大.例如,DARPA 和 Intel 公司合作的 Touchstone 计划[1].自 1984 年欧洲中期天气预报中心(ECMWF)每 2 年召开 1 次并行处理在气象上的应用国际会议[2].美国 nCUBE 公司已研制出具有 8192 个处理机的并行处理系统 nCUBE 2,并将于 1994 年推出具有 64K 个处理机的并行处理系统 nCUBE 3[3].美国的国家计划 HPCC[4]以及欧洲的 ESPRIT 计划[5]均在并行处理上投入很大.

那么,并行处理究竟能有多大的效果呢?欲回答这一问题,需要有针对性地进行具体的分析.Gustafson 基于 scaled speedup[6]对这一问题进行了分析,X. H. Sun 基于 sizeup[7]分析了这一问题.还有很多研究者分别从不同的角度对这一问题进行了很有价值的分析[8,9].本文从应用的角度分析这一问题.本文的组织如下:第 1 部分简单介绍一些预备知识,第 2 部分提出一个性能分析模型,并基于这一模型对影响并行处理性能的诸因素进行详细地分析,最后一部分为结束语.

* 本文 1993-07-28 收到

本论文工作受国家 863 高技术项目 863-306-101 的资助,同时受国家博士点基金(No. 0249136)资助.作者刘德才,31 岁,博士生,目前主要从事于多机调度、并行性控制、并行计算和计算机性能评价的研究工作.王鼎兴,56 岁,教授,目前主要从事于并行处理和智能技术与系统等方面的研究工作.沈美明,56 岁,教授,目前主要从事于并行处理和智能技术与系统等方面的研究工作.郑纬民,48 岁,教授,目前主要从事于并行处理和智能技术与系统等方面的研究工作.

本文通讯联系人:刘德才,北京 100084,清华大学计算机系

1 预备知识

在并行处理中,通常有两种处理方式:一种被称为功能并行(functional parallelism),另一种被称为数据并行(data parallelism).所谓功能并行是指把问题的求解算法划分成若干个能以流水方式运行的部分,并把这些部分分别装入到不同的处理机上以流水方式运行.所以,功能并行也被称为流水并行(pipelining parallelism).这种处理方式,对于能高效地以流水方式互连的多个处理机构成的并行处理系统,其并行处理的效果较好.在这种处理方式中,数据在处理机网上以粗粒度数据流方式进行流水处理.这种方式的通信开销很大,因此,对网络带宽的要求很高,需要精心地设计并行算法的通信结构和控制结构.否则,并行处理的性能不会很好.这种处理方式的优点是对每个处理机的空间要求不高.

所谓数据并行是指把数据划分成若干块分别映象到不同的处理机上,每一处理机运行同样的处理程序对所分派的数据进行处理.大部分并行处理均采用这种处理方式,尤其是对于计算复杂性很高的问题(如流体力学计算、图象处理)进行并行处理.在这种处理方式中,通常,不同的处理机在计算过程中需要进行一定量的通信.因此,在这种并行处理方式中,也需要根据问题的特点设计合理的并行处理算法,以减小处理机间的通信对并行处理性能的影响.

本文研究在 2 维网格(mesh)网互连的多个 Transputer 并行处理系统上数据并行处理方式的性能.

2 性能模型及性能分析

2.1 建立模型

很多应用问题(如数值天气预报、流体力学计算、图象处理等)均具有适合于并行处理的规则几何计算结构.也就是说,这些问题的计算可以方便地被划分成若干个子问题的计算,这些子问题的计算之间通常有一定的关系.例如,在数值天气预报计算中,可把预报的区域均匀地划分成许多网格块分别映象到不同的处理机上.每一处理机计算一个格子块中各点的数值,但格子块边缘点的值的计算是要用到相邻处理机内的点的值的.因此,这需与相邻处理机打交道.在这种并行处理方式下,各处理机分别计算不同区域的点的值,但边缘点的值的计算是需要与相邻处理机进行通信的.每个处理机的计算开销正比于所分派的格子块的大小,而其通信开销则正比于格子块边缘区域的大小.

由于 *speedup* 是并行处理效果的典型表示,因此,本文的性能分析是基于 *speedup* 的.假设应用问题的大小为 $D\times D$,处理机数目为 $P\times P$,每个处理机存储并处理 $n\times n$ 个点,其中 $n=D/P$(假设 D 是 P 的倍数),这样共划分为 $D\times D/(P\times P)$块数据域,相邻数据域被分派到相邻处理机上.每块数据域的边缘 e 宽度的区域中各点值的计算需要用到相邻数据域的边缘点的值(对于许多问题,如图象处理,e 通常为 1).这些边缘点的值是通过处理机间的通信而获得的.如果每个点的计算只用到相邻的四个点的值(如图 1 所示),则

$$DATA_{new}[i,j] = func(DATA_{old}[i-1,j], DATA_{old}[i+1,j], DATA_{old}[i,j-1], DATA_{old}[i,j+1], DATA_{old}[i,j]).$$

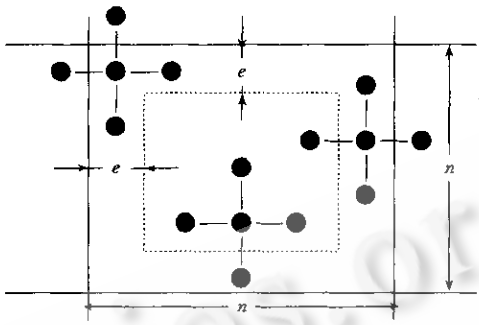

图 1 网格划分及数据点的关系

假设每个数据点的一次计算时间为 T_{cal}，在相邻处理机间每个点一次传输的时间为 T_{com}，每次通信所占用的处理机时间为 T_{set}，并设相邻数据域被分配到相邻的处理机中. 那么，每一计算步每个处理机需要从相邻处理机获得 $4\times e\times n$ 个点的数据，同时需发送 $4\times e\times n$ 个点的数据到相邻的处理机. 如通信和计算无重叠，则每个处理机的总通信时间为 $2\times 4\times e\times n\times T_{com}+2\times 4\times T_{set}$，总计算时间为 $n\times n\times T_{cal}$. 因此，一个计算步的整个并行处理的时间 TP 为 $2\times 4\times e\times n\times T_{com}+2\times 4\times T_{set}+n\times n\times T_{cal}$，所以，加速比 $speedup$ 为：

$$speedup = \frac{D^2 T_{cal}}{8e\frac{D}{P}T_{com} + 8T_{set} + \frac{D^2}{P^2}T_{cal}}.$$

设

$$\mu = \frac{T_{com}}{T_{cal}}, \qquad \nu = \frac{T_{set}}{T_{cal}},$$

则

$$speedup = \frac{D^2P^2}{8eDP\mu + 8P^2\nu + D^2}.$$

由上式可见，$speedup$ 与 T_{cal}、T_{com} 和 T_{set} 的绝对值无关，而与其相对比值 μ 和 ν 有关. 因此，从 $speedup$ 的观点看，并行处理系统和并行算法的设计应着眼于 μ 和 ν，而不是 T_{cal}、T_{com} 和 T_{set}.

2.2 性能分析

本小节我们分析各因素对 $speedup$ 性能指标的影响. 由上一小节的 $speedup$ 公式，我们得：

$$\frac{\partial(speedup)}{\partial P} = \frac{8eD^3P^2\mu + 2D^4P}{(8eDP\mu + 8P^2\nu + D^2)^2}.$$

由于 $e>0$, $D>0$, $P>0$, $\mu>0$ 和 $\nu>0$,因此,

$$8eD^3P^2\mu + 2D^4P > 0,$$
$$(8eDP\mu + 8P^2\nu + D^2)^2 > 0,$$

所以,对于参数 P 来说,*speedup* 没有最大值. 由于

$$\frac{\partial(speedup)}{\partial P} > 0,$$

所以,*speedup* 将随 P 的增大而增大. 那么,*speedup* 能否随着 P 的增大而无穷地增大呢? 当 $P\to\infty$时,$\frac{\partial(speedup)}{\partial P}\to 0$, 并且 $speedup\to D^2/(8\mu+8\nu+1)$*. 因此,虽然随着处理机个数的增加,*speedup* 单调连续地增加,但 *speedup* 有一个可以逼近却不能达到的极限值. 也就是说,存在一个饱和点,当处理机的数目超过此饱和点时,*speedup* 基本上不再增加. 对于其他的参数,可以进行与上类似的分析. 其分析结果列于表 1 中.

表 1　$T_P=8e\frac{D}{P}T_{com}+8T_{set}+\frac{D^2}{P^2}T_{cal}$

$[speedup=D^2P^2/(8eDP\mu+8P^2\nu+D^2)]$

影响性能的因素			*speedup*	
名字	变化	值域	变化	极限值
P	↑	$(0, +\infty)_{integer}$	↑	$D^2/(8\mu+8\nu+1)$
D	↑	$(0, +\infty)_{integer}$	↑	P^2
e	↓	$(0, +\infty)_{integer}$	↑	P^{2**}
μ	↓	$(0, +\infty)_{real}$	↑	$(D^2P^2)/(8P^2\nu+D^2)$
ν	↓	$(0, +\infty)_{real}$	↑	$(DP^2)/(8eP\mu+D)$

现在,我们分析一下操作重叠对并行处理性能的影响. 有三种操作重叠:一种情况是一个处理机的所有通信连路并行重叠地进行数据的传输,第二种是一个处理机的运算操作与连路的通信以及各连路的通信均并行重叠地进行,第三种情况是每一处理机包括一个独立的通信管理硬件,使得通信对于处理机的计算来说是透明的.

对于第 1 种情况,我们有

$$T_P = e\frac{D}{P}T_{com} + 8T_{set} + \frac{D^2}{P^2}T_{cal}.$$

对于第 2 种情况,我们有

$$T_P = 8T_{set} + \max\{e\frac{D}{P}T_{com}, (\frac{D}{P} - 2e)^2T_{cal}\} + 4(e\frac{D}{P} - e^2)T_{cal}.$$

---

* 当 $P^2>D^2$ 时,至少有 P^2-D^2 个处理机处于空闲状态,其他 D^2 个处理机只处理一个数据点.

** 当 $e=0$ 时,处理机间无需通信. 因此,无通信启动时耗,即 $T_{set}=0$. 所以,$T_P=D^2T_{cal}/P^2$.

在第2种情况下，如果每一块数据域的中间$(n-2e)^2$大小的子区域的计算时间$(D/P-2e)^2T_{cal}$均不小于相邻边缘数据点的传输时间eDT_{com}/P，使得在计算完中间子区域时立即进入边缘部分各点的计算. 在这种情况下，有

$$T_P = 8T_{set} + (\frac{D}{P} - 2e)^2T_{cal} + 4(e\frac{D}{P} - e^2)T_{cal} = 8T_{set} + \frac{D^2}{P^2}T_{cal}.$$

可以看出，这种情况相当于没有通信的传输开销.

对于第3种情况，即有单独的硬件部分管理连路的通信，而不中断处理机，那么，处理机的计算可以和通道的通信完全并行地进行. 此时，

$$T_P = \max\{8T_{set} + e\frac{D}{P}T_{com}, (\frac{D}{P} - 2e)^2T_{cal}\} + 4(e\frac{D}{P} - e^2)T_{cal}.$$

在这种情况下，如果每块数据域的中间$(n-2e)^2$大小的子区域的计算时间$(D/P-2e)^2T_{cal}$不小于边缘数据点的通信时间$8T_{set}+eDT_{com}/P$，则可保证处理机无等待地计算. 此时，

$$T_P = (\frac{D}{P} - 2e)^2T_{cal} + 4(e\frac{D}{P} - e^2)T_{cal} = \frac{D^2}{P^2}T_{cal}.$$

这种情况相当于无通信开销，是一种理想的并行处理形式. 此时，

$$speedup \equiv \frac{D^2T_{cal}}{\frac{D^2}{P^2}T_{cal}} = P^2,$$

$$\text{并行处理效率} \equiv \frac{P^2}{P^2} = 100\%.$$

因此，通过操作重叠和合理的数据划分可以获得理想的*speedup*和并行处理效率. 表2—表6分别给出上述5种情况下，各种因素对加速比的影响.

表2 $T_P=e\frac{D}{P}T_{com}+8T_{set}+\frac{D^2}{P^2}T_{cal}$

$[speedup=D^2P^2/(eDP\mu+8P^2\nu+D^2)]$

影响性能的因素			*speedup*	
名字	变化	值域	变化	极限值
P	↑	$(0,+\infty)_{integer}$	↑	$D^2/(\mu+8\nu+1)$
D	↑	$(0,+\infty)_{integer}$	↑	P^2
e	↓	$(0,+\infty)_{integer}$	↑	P^2
μ	↓	$(0,+\infty)_{real}$	↑	$(D^2P^2)/(8P^2\nu+D^2)$
ν	↓	$(0,+\infty)_{real}$	↑	$(DP^2)/(eP\mu+D)$

表 3　$T_P=8T_{set}+\max\{e\frac{D}{P}T_{com},(\frac{D}{P}-2e)^2T_{cal}\}+4(e\frac{D}{P}-e^2)T_{cal}$

$[speedup=D^2P^2/(8P^2\nu+\max\{eDP\mu,(D-2eP)^2\}+4eP(D-eP))]$

影响性能的因素			speedup	
名字	变化	值域	变化	极限值
P	↑	$(0,+\infty)_{integer}$	↑	$D^2/(\mu+8\nu+1)$*
D	↑	$(0,+\infty)_{integer}$	↑	P^2
e	↓	$(0,+\infty)_{integer}$	↑	P^2
μ	↓	$(0,+\infty)_{real}$	↑	$(D^2P^2)/(8P^2\nu+D^2)$
ν	↓	$(0,+\infty)_{real}$	↑	$D^2P/(eD\mu+4eD-4e^2P)$或 P^2

表 4　$T_P=8T_{set}+\frac{D^2}{P^2}T_{cal}$

$[speedup=D^2P^2/(8P^2\nu+D^2)]$

影响性能的因素			speedup	
名字	变化	值域	变化	极限值
P	↑	$(0,+\infty)_{integer}$	↑	$D^2/(\mu+8\nu+1)$
D	↑	$(0,+\infty)_{integer}$	↑	P^2
e	↓	$(0,+\infty)_{integer}$	↑	P^2
μ	↓	$(0,+\infty)_{real}$	↑	$(D^2P^2)/(8P^2\nu+D^2)$
ν	↓	$(0,+\infty)_{real}$	↑	P^2

表 5　$T_P=\max\{8T_{set}+e\frac{D}{P}T_{com},(\frac{D}{P}-2e)^2T_{cal}\}+4(e\frac{D}{P}-e^2)T_{cal}$

$[speedup=D^2P^2/(\max\{8P^2\nu+eDP\mu,(D-2eP)^2\}+4eP(D-eP))]$

影响性能的因素			speedup	
名字	变化	值域	变化	极限值
P	↑	$(0,+\infty)_{integer}$	↑	$D^2/(\mu+8\nu+1)$
D	↑	$(0,+\infty)_{integer}$	↑	P^2
e	↓	$(0,+\infty)_{integer}$	↑	P^2
μ	↓	$(0,+\infty)_{real}$	↑	$D^2P/(8P\nu+4eD-4e^2P)$或 P^2
ν	↓	$(0,+\infty)_{real}$	↑	$D^2P/(eD\mu+4eD-4e^2P)$或 P^2

2.3　小　结

从表 1 至表 6 中可以看出,(1)在并行处理中,处理机的个数应合适.如果处理机个数太少,则并行处理的效果不明显.如果处理机的个数太多,则由于并行性受所处理问题规模的限制,造成过多的处理机浪费(因为 *speedup* 不会超过 $D^2/(\mu+8\nu+1)$).(2)并行处理适合于对大问题的处理.当问题的规模足够大时,在合理的数据划分和通信管理的情况下,并行处理的效果会很理想的(*speedup* 接近于处理机的个数).(3)操作的重叠可以大大提高并行处理的性能.(4)*speedup* 与 T_{cal}、T_{com}和 T_{set}的绝对值无关,而与其相对比值 μ 和 ν 有关.

* 当 $P^2\geqslant D^2$ 时,每个处理机最多只能处理一个数据点.此时,通信和计算无法重叠进行.

表6 $T_P=\frac{D^2}{P^2}T_{cal}$

$$speedup=\begin{cases}P^2 & \text{当 } P^2<(D^2/(\delta_1\delta_2))\text{时}\\ D^2/(\mu+8\nu+1) & \text{当 } P^2\geqslant D^2 \text{ 时}\\ (D^2/(\mu+8\nu+1))\sim P^2 & \text{当 } (D^2/(\delta_1\delta_2))\leqslant P^2<D^2 \text{ 时(其中,}\delta_1\text{ 和 }\delta_2\text{ 是二个正整数)}\end{cases}$$

影响性能的因素			speedup	
名字	变化	值域	变化	极限值
P	↑	$(0,+\infty)_{integer}$	↑	$D^2/(\mu+8\nu+1)$*
D	↑	$(0,+\infty)_{integer}$	↑	P^2(无影响)
e	↓	$(0,+\infty)_{integer}$	↑	P^2(无影响)
μ	↓	$(0,+\infty)_{real}$	↑	P^2(无影响)
ν	↓	$(0,+\infty)_{real}$	↑	P^2(无影响)

尽管前面分析的是一个计算步的情况,但其结果对于任意多次计算步的情况均适用.因为在实际应用中每一计算步的计算过程是相同的,即每一计算步所花费的时间是相同的.设问题计算 M 步,则其串行处理时间 T_{seq} 和并行处理时间 T_{par} 分别为:

$$T_{seq}=M\times D^2\times T_{cal},\ T_{par}=M\times T_p.$$

因此,

$$speedup=\frac{T_{seq}}{T_{par}}=\frac{M\times D^2\times T_{cal}}{M\times T_p}=\frac{D^2\times T_{cal}}{T_p}.$$

看出,这与一个计算步的情况是一样的.

结束语:在并行处理领域中,人们最关心的问题之一是具有 N 个处理机的并行处理系统较单处理机能快多少倍.由于并行处理的效果不但与计算机系统有关,还与问题的求解有关,因此,对于这一问题,需要针对具体的情况进行具体的分析.很多研究者分别从不同的角度对上述问题进行了很有价值的探讨.在本文中,我们从应用的角度分析了这一问题.通过分析,我们认为数据并行处理方式的并行处理效果较好.而且我们得出:在这种处理方式下,(1)$speedup$ 与 T_{cal}、T_{com} 和 T_{set} 的绝对值无关,而与其相对比值 μ 和 ν 有关.(2)并行处理适合于对大问题的处理.当问题的规模足够大时,在合理的数据划分和通信管理的情况下,并行处理的效果会很理想的.(3)操作的重叠可以大大提高并行处理的性能.

参考文献

1 Sigurd L. Lillevik. Touchstone program overview. Proc. of the 5th Distributed Memory Computing Conf., 1990: 647—657.

2 Tuomo Kauranne. A view on massively parallel computing. Computing, 1991;53:16—24.

3 Peter Wusten. Massively parallel systems, the supercomputers for the 90's and beyond. nCUBE Reports, 1992.

4 President's Office of Science and Technology Policy, Grand chal—lenges 1993: High performance computing and

* 当 $P^2\geqslant D^2$ 时,每个处理机最多只能处理一个数据点.此时,通信和计算无法重叠进行.

communications. National Science Foundation, Computer and Information Science and Engineering Directorate, 1800 G St., N. W., Washington, D. C., 20550.

5 Anthony J G Hey. Concurrent supercomputing in Europe. Proc. of the 5th Distributed Memory Computing Conf., 1990:630－646.

6 Gustafson J L. Reevaluating Amdahl's law. Communications of the ACM, 1988;**31**(**5**):532－533.

7 Sun X H *et al*. Toward a better parallel performance metric. Parallel Computing, 1991;**17**(**10&11**):1093－1109.

8 Barton M *et al*. Computing performance as a function of the speed, quantity, and cost of the processors. Proc. Supercomputing'89, 1989:759－764.

9 Liu Decai *et al*. Performance modelling of massively parallel computing in geometric applications. Proc. Int. AMSE Conf. on Modelling, Simulation and Control, 1992(**1**):124－134.

POTENTIAL IN DATA PARALLELISM

Liu Decai, Wang Dingxing, Shen Meiming and Zheng Weimin

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract　This paper, from application viewpoint, analyzes the potential in data parallelism. First, they propose a speedup analysis model. Then, based on the model a comprehensive analysis on speedup is presented. These analytical results could be used to guide the design of parallel processing systems as well as parallel algorithms.

Key words　Parallel processing, performance analysis, speedup.