

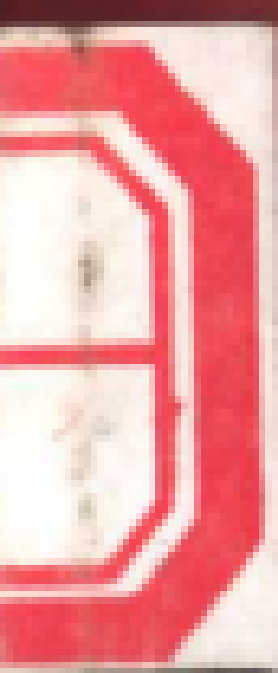
蒙特卡罗方法 及其应用

第四卷

黄流兴 牛胜利 主编

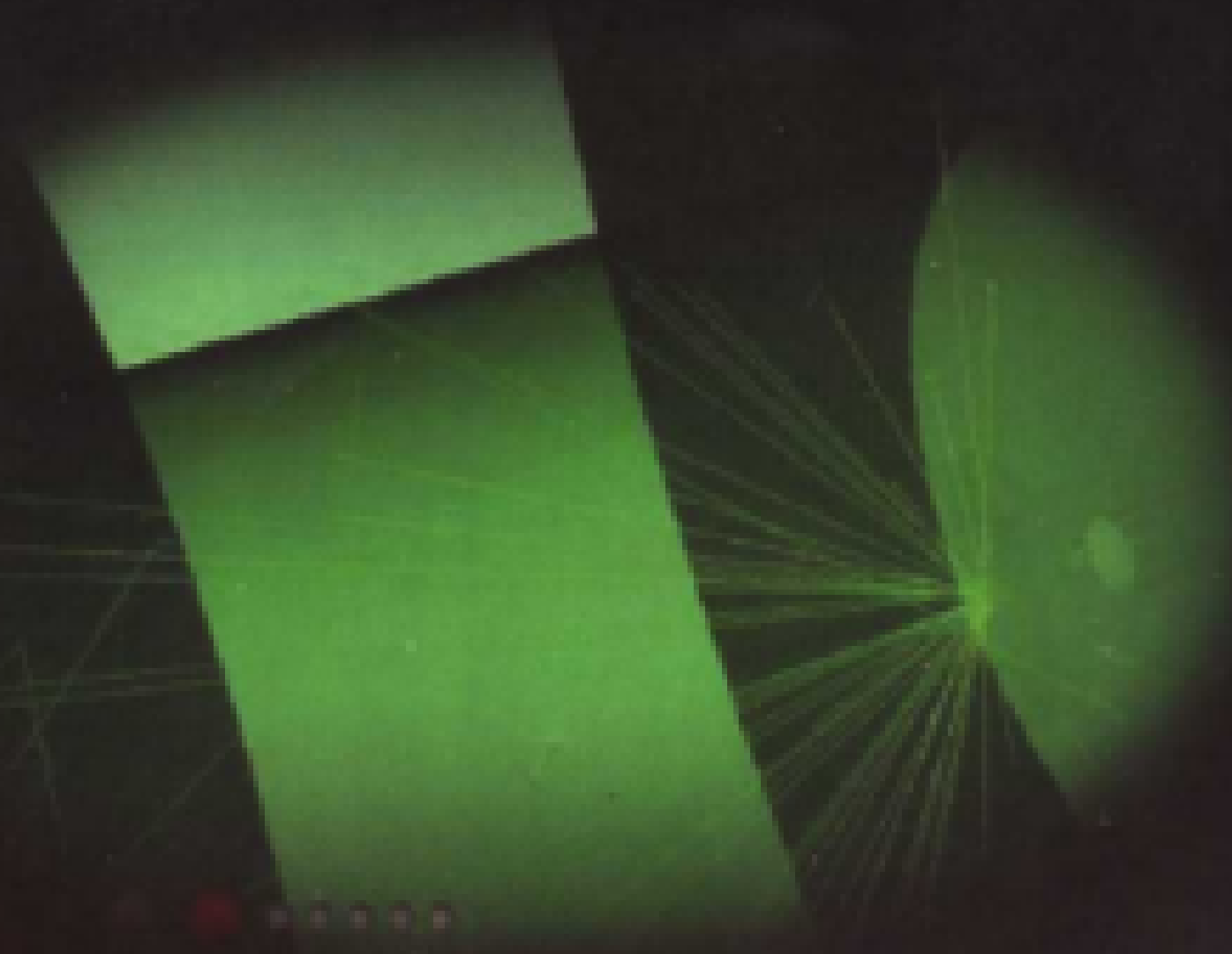


陕西科学技术出版社



MENGTEKALUOFANGFAJIQIYINGYON

责任编辑 屈马珑
封面设计 史文博



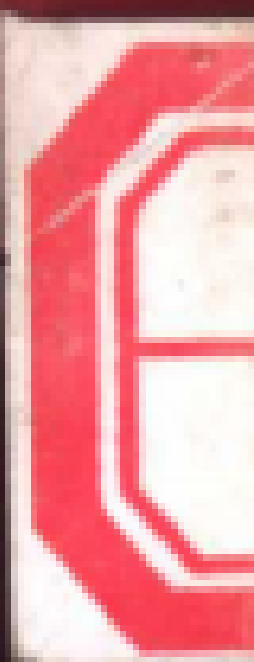
ISBN 7-5369-3816-0



9 787536 938168 >

ISBN 7-5369-3816-0/S·394

定价: 40.00元



MENGTEKALUOFANGFAJIQIYINGYONG

蒙特卡罗方法及其应用

(第四卷)

黄流兴 牛胜利 主编

陕西科学技术出版社

前 言

蒙特卡罗方法是一种经典的数学方法。随着计算机技术的发展,该方法以其独特的优势在核科学、统计物理、分子动力学、医学等自然科学领域得到广泛应用,同时在金融、风险分析等方面也得到了应用。我国在计算物理学会设立了蒙特卡罗方法及其应用专业委员会,定期组织召开全国蒙特卡罗方法及其应用学术交流会,并不定期举办学术讲座和研讨班。自1980年以来,已成功举办了八届全国蒙特卡罗方法及其应用学术交流会。早先几届会议交流的论文没有正式出版。第5、6、7届全国蒙特卡罗学术会议论文集在出版社正式出版,如1993年、1997年和2000年分别由国防科技大学出版社、海洋出版社和安徽大学学报(自然科学版)出版的论文集,对我国该领域的发展起到了积极的作用。

第八届全国蒙特卡罗方法及其应用学术交流会由西北核技术研究所承办,于2004年3月28日至4月2日在云南昆明成功召开。会议交流了近年来蒙特卡罗方法和应用方面的研究成果,并研讨了蒙特卡罗方法在我国的发展方向。会议论文经蒙特卡罗专业委员会委员评审,对部分文章提出了修改意见,录用的论文结集为本书,并承接以前的书名,以多卷本丛书系列之第四卷形式出版。内容反映了近年来我国在蒙特卡罗方法及其应用方面的研究成果和水平,相信将对我国蒙特卡罗方法及其应用领域和相关学科的发展起到良好的推动作用。

本书的出版得到了西北核技术研究所和陕西科学技术出版社的大力支持,专业组裴鹿成、王仲奇、邓力、王瑞宏、李君利、包景东等专家审阅了全书稿件,西北核技术研究所赵军、朱金辉、彭玉、杨斌、庞天燕为此书的出版付出了大量辛勤的劳动,在此表示诚挚的感谢。

黄流兴 牛胜利

2004年4月于西安

目 录

第一篇 蒙特卡罗方法与程序

非定常粒子输运蒙特卡罗并行算法研究	邓 力等/3
点通量替代方法在解小探测器通量计算中的应用	王朝辉等/12
蒙特卡罗研究中的一种几何变换方法	赵 军等/17
分裂与轮盘赌方法在解深穿透问题中的应用	郑玉来/21
求解输运问题的自适应蒙特卡罗重要抽样方法	上官丹骅/26
EGSWIN 系统的完善与使用	邱 睿等/33
蒙特卡罗程序 MONK-8B 简介	王维善等/38
通用蒙特卡罗程序 NMCP1.0 介绍之一	范佳锦等/47
通用蒙特卡罗程序 NMCP1.0 介绍之二	范佳锦等/51
MCNP 输入卡文件可视化编辑程序开发	杨平利等/55

第二篇 蒙特卡罗方法在核技术中的应用

低本底反康谱仪的蒙特卡罗模拟	曾 志等/63
利用 MCNP 程序计算晶体的 n 、 γ 分辨能力	宋朝晖/69
电子加速器中的光中子模拟计算	武 祯等/77
蒙特卡罗模拟近似计算 AWCC 中子符合计数效率	赵学军等/85
LSO 闪烁探测器灵敏度蒙特卡罗计算研究	牛胜利等/91
^{152}Eu 面源的 γ 射线能谱模拟计算	胡广春等/96
14MeV 中子屏蔽优化设计的蒙特卡罗模拟	赵 军等/101
高 Z 材料中 X 光散射规律的数值研究	施将君等/106
硬 X 射线剂量增强、封装屏蔽的蒙特卡罗模拟计算	郭红霞等/111
计算 γ 定向发射强度的蒙特卡罗方法	王瑞宏等/118
用蒙特卡罗法研究 10MeV 量级电子束打靶产生韧致辐射 X 光的规律 ..	龙继东等/125
移动中子照相系统中子源正交优化设计	屠 荆等/131
高能闪光照相中有效吸收系数的蒙特卡罗计算	刘 进等/136
针孔成像系统中定向出射强度的蒙特卡罗计算	朱金辉等/143
硼中子俘获治疗脑癌的蒙特卡罗模拟	邱有恒等/149

用蒙特卡罗方法计算光子外照射对人体产生的有效剂量	王丰春等/156
数字化乳腺 X 射线摄影优化能谱的数值模拟研究	魏福利等/162
壳层增殖系统水淹条件下的临界安全分析	付广智等/171
铀氢锆脉冲堆蒙特卡罗模拟计算	陈伟等/178
用蒙特卡罗方法进行核临界实验数据的验证计算和实际问题的计算	荆永宇等/184
基于蒙特卡罗方法的 MCNP 程序在 CARR 核发热计算的应用	孙志勇等/194
脉冲堆快中子辐射场的蒙特卡罗模拟	江新标等/198
MCNP 程序在微堆低浓化改造方案设计中的应用	孙志勇等/204
Monte Carlo Simulation for X-ray Imaging	宁静等/208
The Status of Monte Carlo Simulation in Japan	杨锦安/216
利用蒙特卡罗方法对中子瞬发伽玛煤炭成分分析系统进行初步设计	贾文宝/223
LiD 中子物理的蒙特卡罗模拟计算	许淑艳等/227
局域中空光束中原子强度梯度冷却的蒙特卡罗研究	印建平/231

第三篇 蒙特卡罗方法在统计物理及其他方面的应用

周期场中定向 Lévy 飞行的蒙特卡罗模拟	王海燕等/239
二维布朗动力学的蒙特卡罗模拟	赵同军等/243
离子通道的通透性机制的蒙特卡罗模拟	安海龙等/247
碳纳米管储氢的物理吸附机制	袁兴红等/251
单壁碳纳米管储氢的 GCMC 模拟	张立波等/256
多壁碳纳米管储氢的蒙特卡罗模拟	唐瑞华等/260
多孔介质中气体扩散的蒙特卡罗模拟	朱金辉等/265
蒙特卡罗方法在水动力学中的应用研究	吉庆丰等/270
CTBT 全球惰性气体监测网监测能力评估方法研究	邝飞虹等/276
放电气体温度对快原子态氮(N^+ , N_f)的影响	张连珠等/281

第 一 篇

蒙特卡罗方法

与 程 序

非定常粒子输运蒙特卡罗并行算法研究

邓力¹ 张文勇² 刘杰² 许海燕¹ 王瑞宏¹ 李树¹

(1. 北京应用物理与计算数学研究所 北京 100088; 2. 国防科技大学计算机学院 长沙 410073)

摘 要 本文针对非定常粒子输运蒙特卡罗并行计算,研究给出了包括自适应并行、MPI 聚合并行和 Master-Slave 并行等多种算法,特别是散射源分层抽样和期望估计,较好地解决了制约并行可扩展性的瓶颈,新算法与串行结果相符,加速比呈线性增长。

关键词 自适应并行, MPI 聚合并行, Master-Slave 并行, 散射源分层抽样, 期望估计

引言

蒙特卡罗方法(下简记为 MC)应用于粒子输运求解已是众所周知的事,然而由于 MC 的误差与跟踪的样本数的平方根成反比,通常要达到误差要求,需要模拟大量粒子历史,从而花费大量的计算时间。特别对非定常粒子输运问题的 MC 模拟,需要模拟上千个时间步,每步需要跟踪上万个粒子,对每个时间步其等价于模拟一个定常问题。由于粒子寿命远超过单个时间步长,因此,当前时间步剩余粒子状态量的保存,以及下个时间步剩余粒子的继续跟踪形成一个循环。这是非定常问题不同于定常问题的主要之处,由于每步涉及大量数据的通讯, I/O 并行是问题之关键。

非定常问题由多个时间步组成,每个时间步对应一个定常计算。虽然多时间步问题串行计算时间比定常问题长得多,但每步的计算时间和粒度都不大,相比定常问题, I/O 比重明显加大。原则上定常并行取得的成果可以移植到非定常计算中,而并行化的主要难点是在计算粒度很有限、I/O 比重较大情况下,如何实现高效并行。现在的计算机, I/O 速度大大落后于处理器的速度,这就更加显示了解决 I/O 问题的重要性,我们的许多工作就是围绕着 I/O 进行的。研究了两类多时间步粒子输运问题:大步长问题和小步长问题。对大步长问题,由于每步串行计算时间差异较大(几分钟到几十分钟),采用相同处理器从始到终计算会产生两种效果:(1)某些步处理器不足;(2)某些步处理器过多,并行效率不增反降。针对此种现象,本文给出了自适应并行处理器选取方法。对小时间步问题,计算稳定性较好,但每步的计算粒度有限, I/O 比重大,针对这种问题,我们发展了几种并行 I/O 算法,为了减少数据的通讯,我们尽可能地利用内存,把串行程序中的硬盘文件访问转换为内存访问,原来每步退出系统的做法显然不便做到这一点,为此,我们将程序设计为 Master-Slave 模式,即处理器的申请只在初始步,期间时间步的切换均不退出并行区,为公用变量驻留在内存中和高速缓存 Cache 的使用创造了条件。

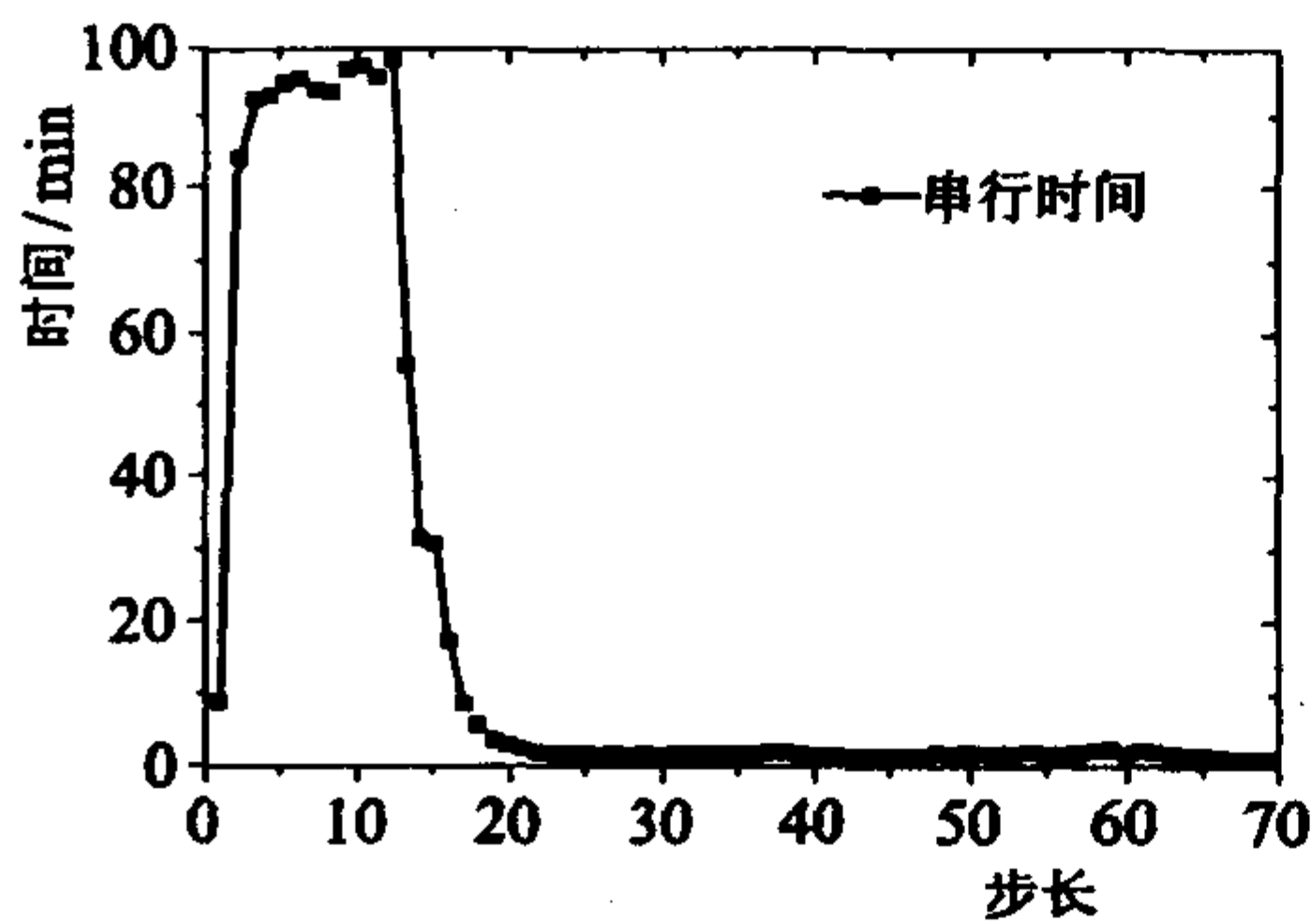
由于国产并行计算机的 I/O 技术远落后进口机,对 I/O 要求很高的非定常粒子输运 MC 程序,如何在国产并行机发挥作用成为本研究的重要内容。本研究针对国产并行机,研究了三种并行 I/O 算法,分别是:(1)自适应并行;(2)MPI 聚合并行;(3) Master-Slave 并行。为确保并行与串行一致的结果,研究同时给出了分段随机数产生法。

1 问题特点及算法研究

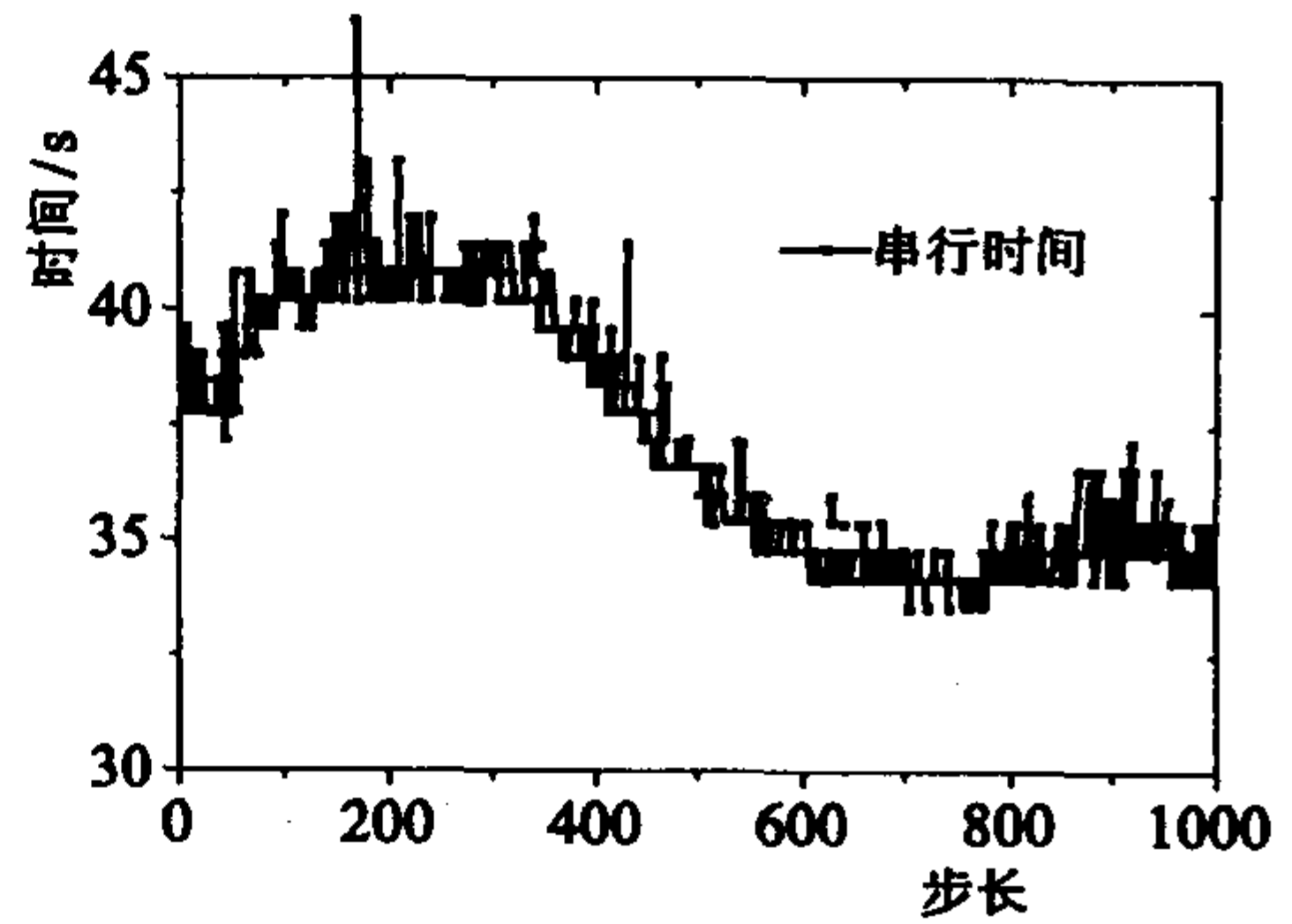
MC 定常粒子输运计算是多处理器并行计算的最佳对象,因为它满足并行计算的基本要

素:(1)数据独立;(2)负载平衡;(3)计算粒度大;(4)I/O 通讯少。因此,问题可以获得高加速比^[1]。但对多时间步粒子输运 MC 计算,条件(3)、(4)不满足。

为了说明计算问题特点,图 1(a)、(b)分别给出大步长问题和小步长问题的串行计算时间分布。从图 1(a)中可以看出,对大步长问题,前若干步计算粒度大(2~99min),这一部分可以获得高加速比,而之后的若干步计算粒度减小。从图 1(b)中可以看出,对小步长问题,每步计算时间十分有限(32~45s),因此它使用的处理器也是非常有限的。两种问题具有不同的特点,因此,采用的计算方法也不尽相同。



(a)



(b)

图 1 多时间步粒子输运问题串行计算时间分布

1.1 大步长问题的自适应并行

处理器多少的选取主要基于 I/O 通讯强度(O_c)和计算强度(T)之间的比例,即 O_c/T ^[2], 这里 O_c 是包括几何网格、截面数据、计数、文件交换及输出的总费用。由于除散射源外的每步通讯强度是基本不变的,因此, O_c 直接由散射源大小决定。根据计算经验,当 $O_c/T \leq 15\%$ 时,处理器增加;当 $O_c/T > R_{th}$ 时(R_{th} 为一阈值,例如取 $R_{th} = 25\%$),处理器减少。

用 N_{proc} 和 $N_{proc-next}$ 分别表示当前步和下一步的处理器数; N_{add} 和 N_{red} 表示处理器的增加数和减少数。则 $N_{proc-next}$ 定义如下:

$$\text{if } (O_c/T \leq 0.15) \quad N_{proc-next} = N_{proc} + N_{add}$$

$$\text{if } (O_c/T > 0.15 \text{ and } O_c/T \leq 0.25) \quad N_{proc-next} = N_{proc}$$

$$\text{if } (O_c/T > 0.25) \quad N_{proc-next} = N_{proc} - N_{red}$$

N_{red} 的选取基于下列条件:

$$\text{if } (N_{proc} \leq 16) \quad N_{red} = 4$$

$$\text{if } (N_{proc} > 16 \text{ and } N_{proc} \leq 32) \quad N_{red} = 8$$

$$\text{if } (N_{proc} > 32 \text{ and } N_{proc} \leq 64) \quad N_{red} = 16$$

$$\text{if } (N_{proc} > 64) \quad N_{red} = 24$$

N_{add} 的选取基于下列条件:

$$\text{if } (N_{proc} \leq 8) \quad N_{add} = N_{proc}$$

$$\text{if } (N_{proc} > 8 \text{ and } N_{proc} \leq 64) \quad N_{add} = 16$$

$$\text{if } (N_{proc} > 64) \quad N_{add} = 24$$

以上处理器的自适应选取依据主要基于经验和计算问题特点。

1.2 并行随机数产生

MC 程序并行化的关键技术之一是随机数的并行。串行随机数产生方法较多,计算机产生的伪随机数序列均匀性较好,但受计算机字长限制,周期有限。为了获得长周期的随机数,采用组合随机数,即由一维随机数向多维随机数发展,由此扩大伪随机数周期。这里给出一维并行随机数产生方式。目前并行随机数发生主要有两种方法,分别为:

1.2.1 跳跃法^[3]

串行伪随机数序列 $\{x_n\}_{n=0-\infty}$, 满足乘同余法, 其递推公式为

$$\begin{cases} x_{n+1} \equiv \lambda x_n \pmod{M}, n = 0, 1, 2, \dots \\ r_{n+1} = x_{n+1}/M, 0 < r_{n+1} \leq 1 \end{cases} \quad (1)$$

其中: λ 为乘子, M 为模, 两者均取素数以避免相关性; x_0 为初始随机数, r_{n+1} 为 $(0, 1)$ 上均匀分布的随机数。

模与计算机字长有关, 它决定了伪随机数的周期, 单精度模可取 $2^{31} - 1$ 。

并行随机数序列 $\{y_n\}_{n=0-\infty}$, 满足递推公式:

$$y_{n+1}^p \equiv \lambda^p y_n^{(p)} \pmod{M}, p = 1, 2, \dots, P; n = 1, 2, \dots \quad (2)$$

其中: P 为并行计算机处理器总数。

直接计算 $\lambda^p y_n^{(p)}$ 可能导致上溢, 利用关系:

$$\alpha^p \pmod{M} \equiv (\alpha^{p-1} \pmod{M}) \alpha \pmod{M} \quad (3)$$

容易快速算出每个处理器的初始随机数。最后求得并行随机数序列 $\{y_n^{(p)}\}$ 与串行随机数序列 $\{x_n\}$ 之间满足关系:

$$\{y_n^{(p)}\} = x_{P(n-1)+p}, n = 1, 2, \dots; p = 1, 2, \dots, P \quad (4)$$

跳跃法: 各处理器使用的随机数序列为:

处理器 1: $x_1, x_{1+p}, x_{1+2p}, x_{1+3p}, \dots$;

处理器 2: $x_2, x_{2+p}, x_{2+2p}, x_{2+3p}, \dots$;

处理器 3: $x_3, x_{3+p}, x_{3+2p}, x_{3+3p}, \dots$;

$\vdots$

处理器 P : $x_p, x_{2p}, x_{3p}, x_{4p}, \dots$ 。

在对定常 MCNP 程序的并行中, 我们较早采用的是这种方法, 目前已改为分段法。

1.2.2 分段法

为了取得与串行程序一致的结果, 采用与串行程序相同的随机数序列是必要的。MCNP 程序每个粒子可用随机数为 $n=4297$ 个, 假定每个处理器使用 m 个随机数, 即第一个粒子使用的随机数范围为 $1 \sim 4297$, 第二个粒子为 $4298 \sim 8594$, 第 m 个粒子为 $n(m-1)+1 \sim mn$ 。并行“分段”随机数序列为:

分段法: 各处理器使用的随机数序列为:

处理器 1: $x_1, x_2, \dots, x_{mn}$;

处理器 2: $x_{mn+1}, x_{mn+2}, \dots, x_{2mn}$;

处理器 3: $x_{2mn+1}, x_{2mn+2}, \dots, x_{3mn}$;

$\vdots$

处理器 P : $x_{(P-1)nm+1}, x_{(P-1)nm+2}, \dots, x_{Pnm}$ 。

使用分段法,需要快速产生每个处理器的初始随机数。由于并行计算所用随机数序列与串行随机数序列一致,从而确保了计算结果的一致性。当并行与串行结果不一致时,可以断定是并行程序设计存在的问题,从而减少了不确定性。

1.3 方法 1——MPI 聚合并行^[3,4]

由于每台处理器跟踪每个粒子状态的计算过程不相关,因此,散射源数据的读、写也是不相关的。对于读散射源,采用并发读取同一个文件方式;对于写散射源,每个处理器独立写各自的局部散射源文件,每步 MC 结束后,将局部文件合并为一个完整文件,供下步使用,这个合并过程由串行完成,图 2 给出了上述过程示意图。

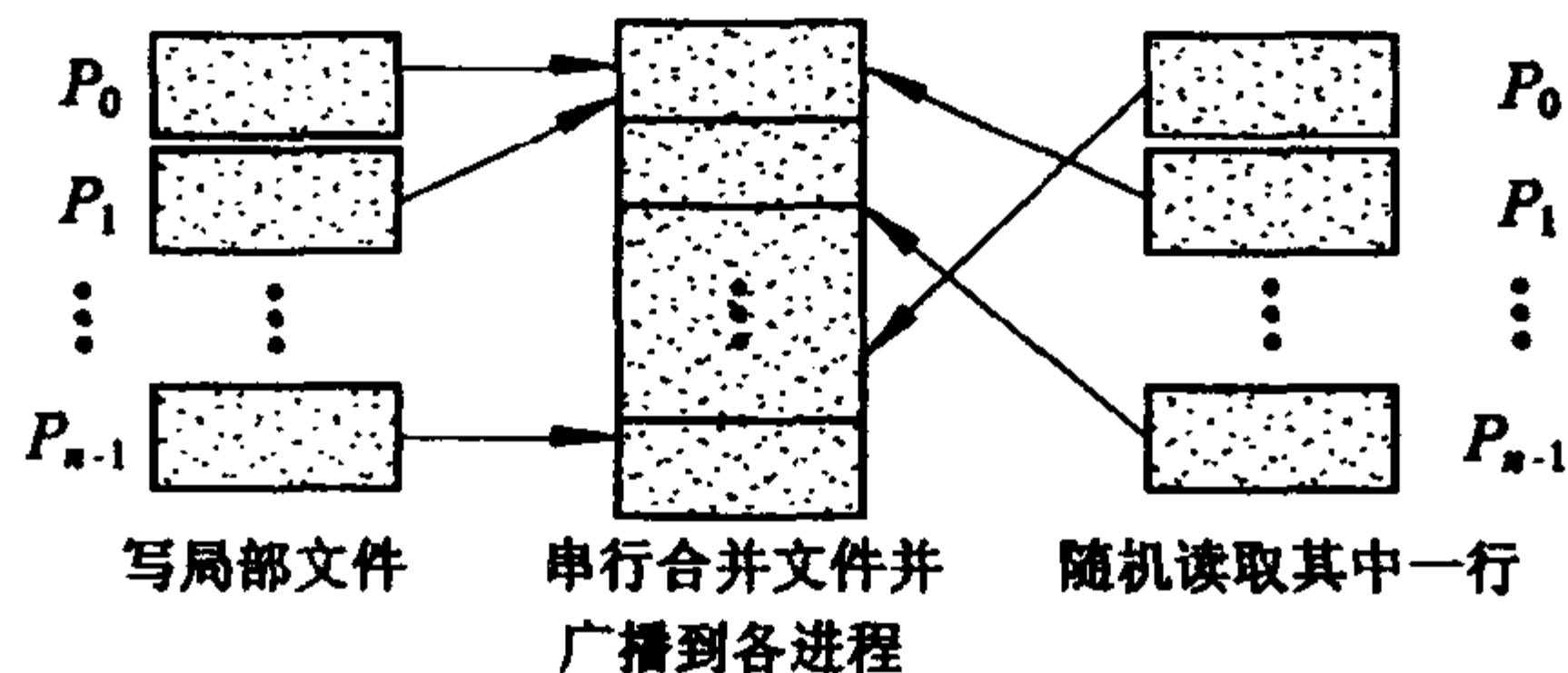


图 2 方法 1 并行 I/O 示意图

MPI2.0 提供了两类 MPI 并行 I/O 函数,一类是每个进程并行访问不同文件;另一类是各个进程并行访问同一文件。因此,方法 1 可以直接调用 MPI2.0 提供的并行 I/O 函数完成,其调用形式为:MPI_file_Open, MPI_File_Read, MPI_File_Write 和 MPI_File_Close。

这种方法程序设计简单,只需将原来的 Open, Read, Write 和 Close 与 MPI2.0 提供的并行 I/O 函数进行相应的修改即可。这种并行 I/O 存取模式由大量的数量很小、但非连续的小数据组成,而一个应用问题如果需要很多小而频繁的 I/O 请求,必然导致 I/O 性能下降,另一方面这些 I/O 操作依赖于每个粒子的跟踪过程,很难利用 MPI2.0 提供的并行 I/O 函数获得高性能。

1.4 方法 2——变硬盘数据访问为内存访问

对于当前并行机系统,从磁盘中存取文件的速度远比从内存中存取数据的速度慢一万到十万倍。另一方面,计算机具有丰富的内存资源,假设存取的数据量一样,那么存取大块数据的时间要比存取小块不连续的数据的时间快得多。

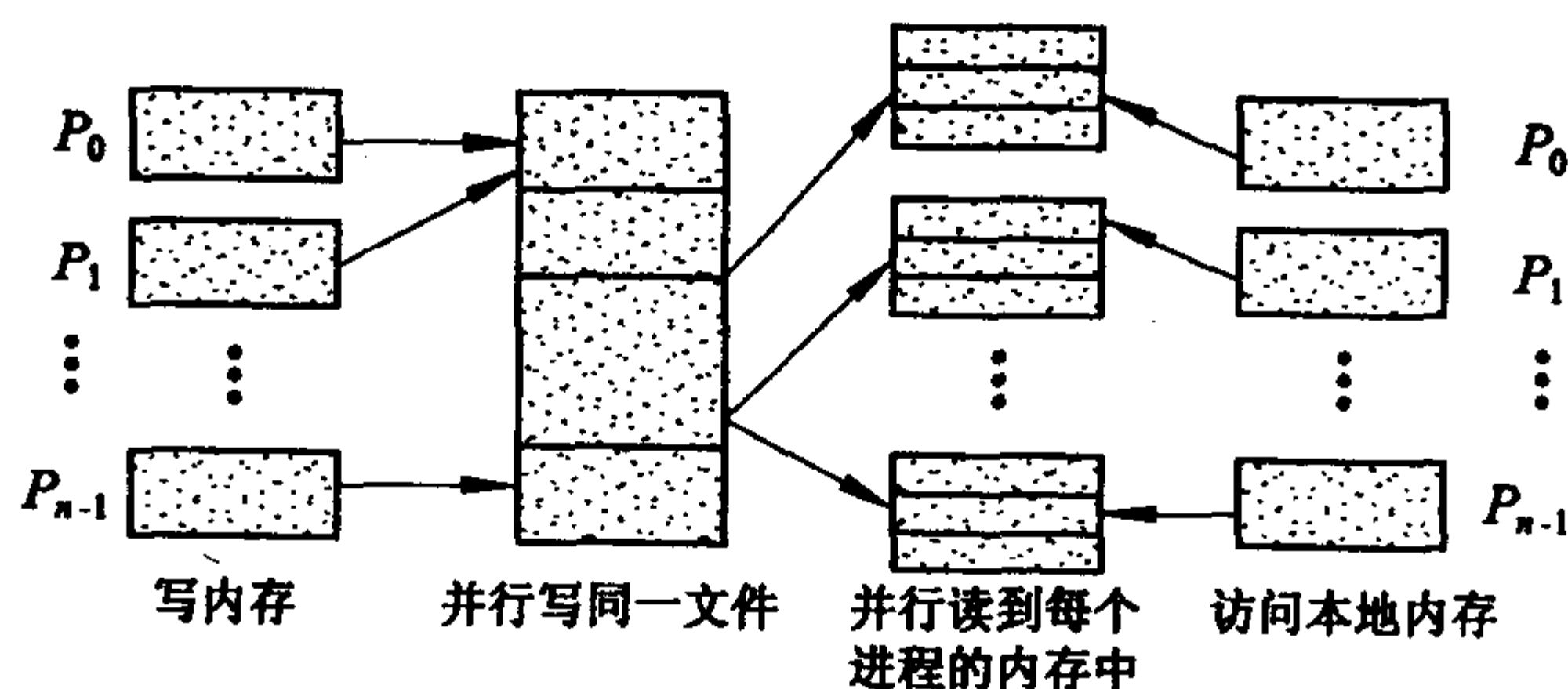


图 3 方法 2 并行 I/O 示意图

基于以上原因,将散射源文件的整个数据一次读到内存中,散射源数据的读取从内存数组中直接获得,每步结束时,将本步生成的新散射源数组一次性写到一个文件中,图3给出了示意图。

将小块数据合成大块数据一起并行写,而将少量、不连续的随机访问散射源文件转化为内存访问,并行性能较方法1有显著改善。

这种方法充分利用了内存和并行 Cache,节约了硬盘资源,对并行 I/O 性能要求不高,国产、进口并行机均适用。

1.5 方法3——Master-Slave 模式

方法1和方法2都是把上个时间步产生的散射源文件写到磁盘上,然后在下个时间步再读取该文件,这两种算法都需要与磁盘打交道,且每个时间步涉及处理器的再申请,不同的计算机申请处理器均有一个延迟时间,进口并行机延迟时间较短(1~2s),且处理器采用共享模式,不存在处理器申请不够的问题(采用分时系统)。然而国产并行机的延迟时间较长(约10~15s),若每步申请处理器的等待时间为10s,对上万步迭代的非定常问题,这是难于接受的。

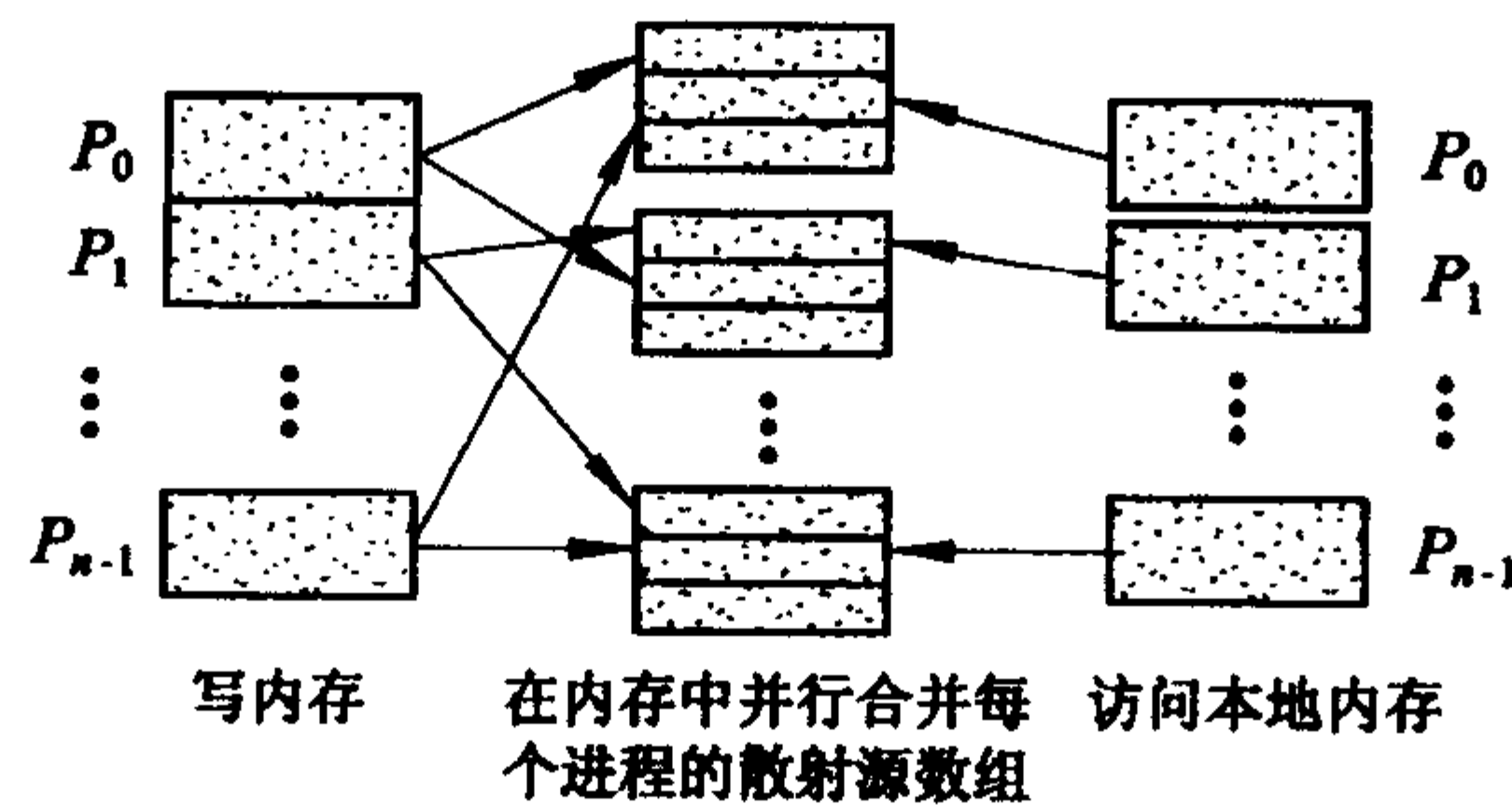


图4 方法3并行 I/O 示意图

因此,对并行程序设计要求有很高的适应性。以方法2为基础,设计以总控程序为主程序,MC程序为子程序的“Master”“Slave”模式(如图4),取代原来的两个对等主程序模式。这一改进不仅克服了上面提到的不足,而且,相对方法2减少一次内存数组与硬盘文件的访问,由于原来的 I/O 磁盘数据文件模式改为内存数组模式,计算效率提高数倍,原来每步需要广播的数据量在此模式下大大减少,如截面数据,只需第一步广播。但这种模式,需要确定局部与全局变量,每步需要初始相应的模块数据。

方法1~3,并行取得了与串行完全一致的结果。方法2、方法3同样适合其他并行机,且并行效果显著。

2 分层抽样与期望估计

2.1 直接估计

假定粒子当前的积存权为 w , $E_1, E_2, \dots, E_n$ 为 n 个相互独立的随机事件,每个事件发生的概率分别为 $p_1, p_2, \dots, p_n$, 满足 $\sum_{i=1}^n p_i = 1$ 。任意抽随机数 $\xi \in [0, 1]$, 求出满足

$$\sum_{i=1}^{j-1} p_i \leq \xi < \sum_{i=1}^j p_i, 1 < j \leq n \quad (5)$$

的 j , 那么 ξ 决定了事件 E_j 的发生, 粒子权不变, 称此随机过程为直接估计。

直接估计有时存在随机性对计算结果的影响。

2.2 期望值估计

把粒子权 w 按每个事件发生的概率, 分配到每个事件上, 其权重为

$$w_i = p_i w, i = 1, 2, \dots, n \quad (6)$$

由此确定了 n 个相互独立事件的游动权, 它没有随机性的影响。这种通过减小权来确保每个事件发生的做法称为期望估计^[5]。

由于少一次抽样, 它的方差小于直接估计, 可避免小概率事件因抽样而被忽略的情况发生, 不过这种做法会增加计算量, 适合并行计算, 例如用 n 个处理器, 每个处理器对应一个事件的处理。

2.3 分层抽样

考虑下列积分 $I = \int_D f(x)g(x)dx$ 的计算, 其中 $f(x)$ 为 D 上的概率密度函数, 若 $D = \bigcup_{i=1}^m D_i, D_i \cap D_{j \neq i} = \phi$, 令 $p_i = \int_{D_i} f(x)dx$ ($\sum_{i=1}^m p_i = 1$), 定义 D_i 上的概率密度函数

$$f_i(x) = \begin{cases} \frac{f(x)}{p_i}, & x \in D_i \\ 0, & x \notin D_i \end{cases}$$

则 I 可以写为 $I = \sum_{i=1}^m p_i \int_{D_i} g(x)f_i(x)dx = \sum_{i=1}^m p_i I_i$, 对每个 i , 用平均值法去估计 I_i , 从而得到 I 的估计量: $\hat{I} = \sum_{i=1}^m p_i \hat{I}_i$ 。首先确定不同源的源强, 设外源源强为 e_0 , 散射源源强为 e_s 。则有

$$e_s = \frac{\bar{e}_s \cdot ws}{N} \quad (7)$$

$$e_0 = 1 - e_s \quad (8)$$

其中: $\bar{e}_s$ 为上一时间步的散射源总源强, N 为源粒子总数, ws 为散射源总权重。

定义散射源事件 A 和外源事件 B , 则事件 A 和事件 B 的概率分别为

$$P(A) = \frac{e_s}{e_s + e_0} \quad (9)$$

$$P(B) = 1 - P(A) \quad (10)$$

MC 并行计算基于将跟踪的 N 个粒子, 等分到 P 个处理器 (假定 N 整除 P), 每个处理器跟踪的粒子数为 $N_0 = N/P$, 则从总体中抽取的散射源粒子总数为 $N \cdot P(A)$, 外源粒子数为 $N \cdot P(B)$ 。若把它们等概率分到 P 个处理器跟踪, 则每个处理器的散射源权重应等于平均权重

$$ws_i = ws_{avg} = \frac{ws}{P}, i = 1, 2, \dots, P \quad (11)$$

若散射源粒子权按归一处理, 则平均到每个处理器的散射源粒子数为

$$N_{0,s} = \text{int}(ws_{avg} + \xi), \xi \in [0, 1] \quad (12)$$

若采用上一时间步每个处理器产生的散射源粒子数, 则每个处理器的实际散射源粒子数为

$N_0 \cdot P_i(A), i=1, 2, \dots, P$, 其中

$$P_i(A) = P(A) \frac{ws_i / nps_0}{ws / nps} = P(A) \frac{ws_i}{ws} P \quad (13)$$

为第 i 个处理器的散射源事件概率, ws_i 为第 i 个处理器的散射源权重。

上式两端同乘以 N_0 , 并对 i 求和, 有

$$\sum_{i=1}^P N_0 \cdot P_i(A) = N \cdot P(A) \quad (14)$$

相应第 i 个处理器的外源事件概率为 $P_i(B) = 1 - P_i(A)$, 外源粒子数为 $N_0 \cdot P_i(B)$ 。于是得到如下散射源权重分解式

$$ws_i = ws \cdot p_i, i = 1, 2, \dots, P \quad (15)$$

其中 $p_i = ws_i / ws$, 满足 $\sum_{i=1}^P p_i = 1$ 。

采用期望估计, 让每个处理器的散射源事件按实际概率 $p_i (i=1, 2, \dots, P)$ 和权重 ws_i 同时发生, 并行跟踪散射源粒子和外源粒子。同时利用分层抽样的做法, 从每个处理器的散射源样本点空间产生散射源分支。

运用串行散射源等概率分组的做法, 并行计算每个处理器的散射源为一组, 共 P 组 (P 为并行计算使用的处理器数), 若依据每个组的自身的散射源分支和权重, 每个组的散射源概率为 $p_i = ws_i / ws$, 通常情况下 $p_i \neq 1/P$, 即 $p_i(A) \neq P(A)$ 。

为了保证计算结果的无偏, 必要的“纠偏”措施需要。如图 5(a), 将超过平均权 ws_{avg} 处理器的多余散射源权重及分支, 迁移到低于平均权的处理器的散射源中, 最终使每个处理器的散射源权重接近平均权 (如图 5(b))。

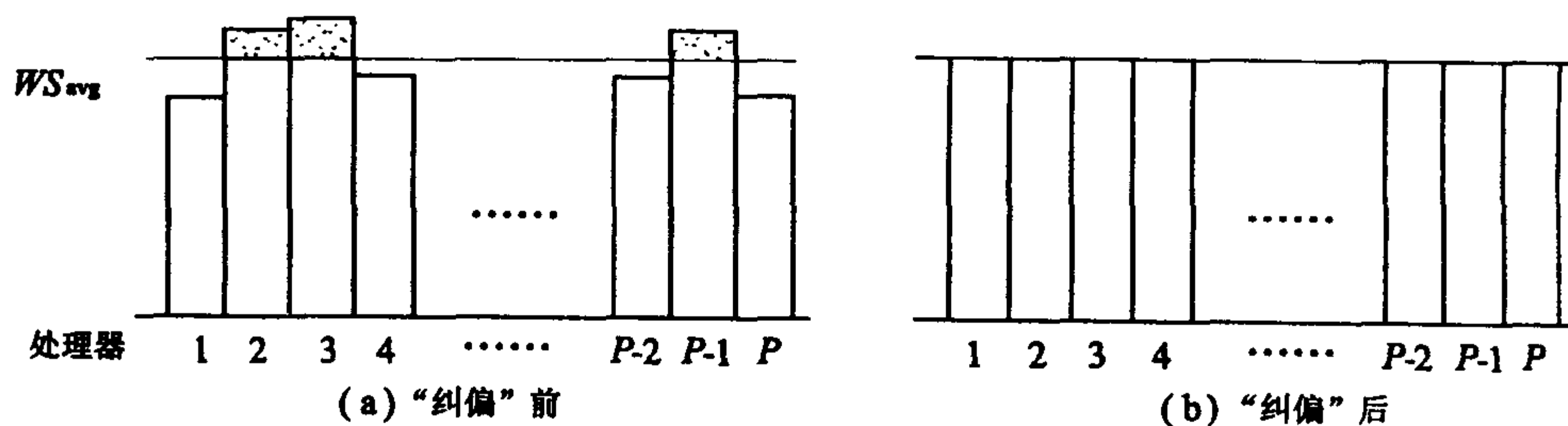


图 5 散射源迁移示意图

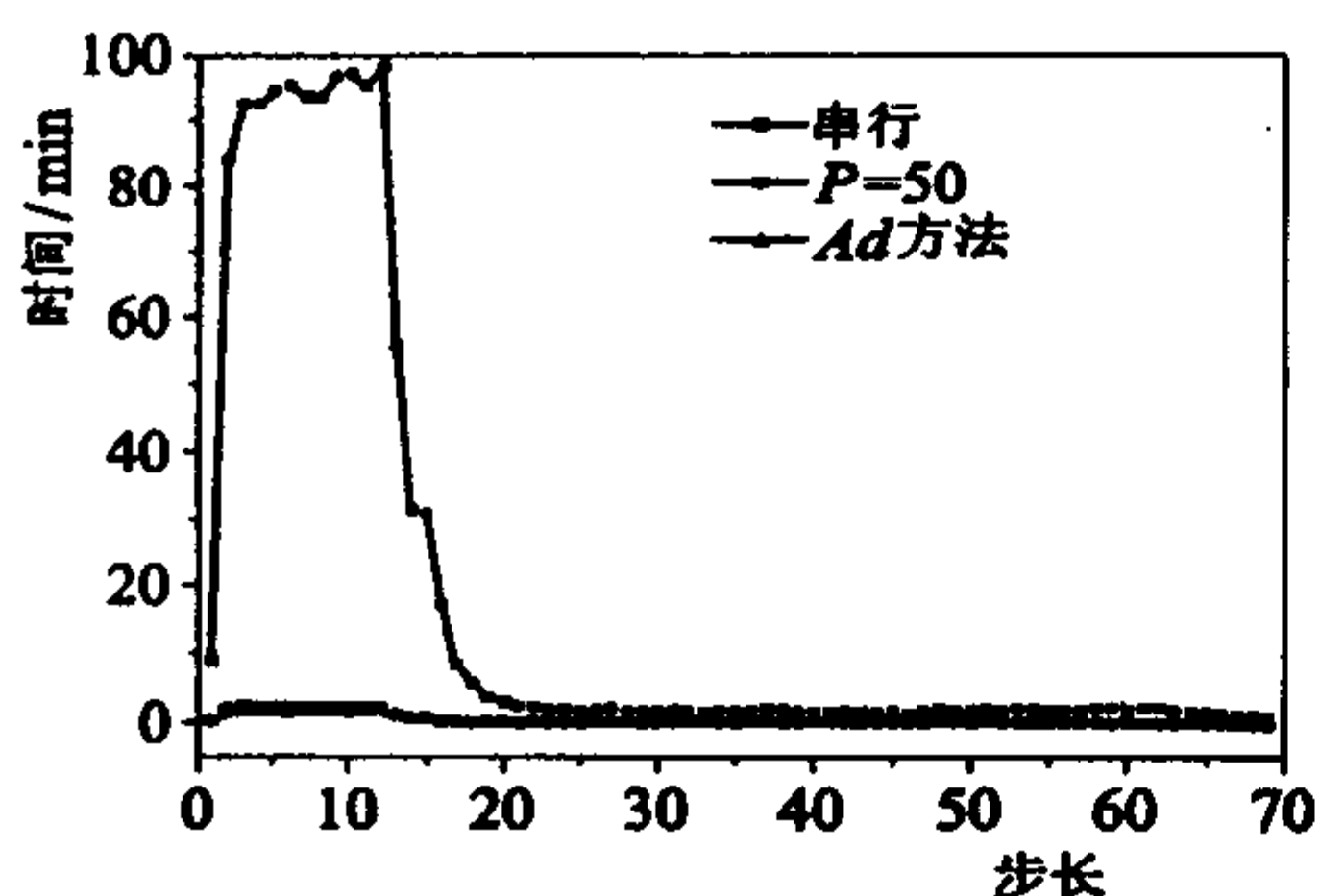
3 计算结果及加速比

表 1 给出了不同处理器及自适应处理器的加速比比较, 以图 1 中的两个问题为并行测试对象, 图 6 给出了大步长问题的自适应并行时间分布。图 7 给出了小步长问题不同算法的加速比比较。

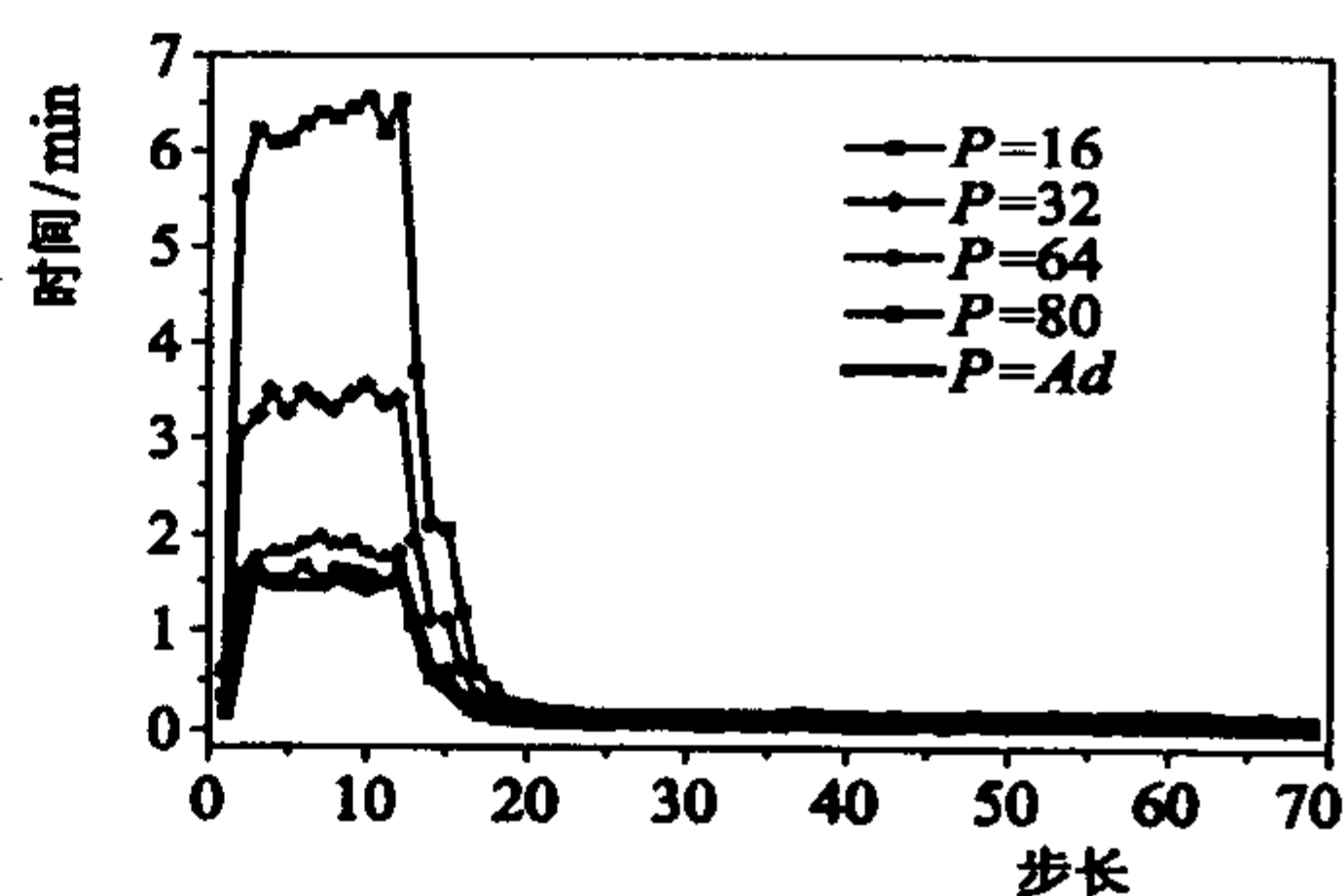
表 1 大步长模型计算时间及加速比

处理器	1	4	8	16	32	64	80	Ad
$S(P)$	1	3.93	7.89	14.85	27.21	48.57	55.76	56.71

注: Ad 表示自适应。



(a) 串行与并行计算时间比较



(b) 不同处理器的计算时间比较

图 6 大步长问题串行与并行计算时间对比

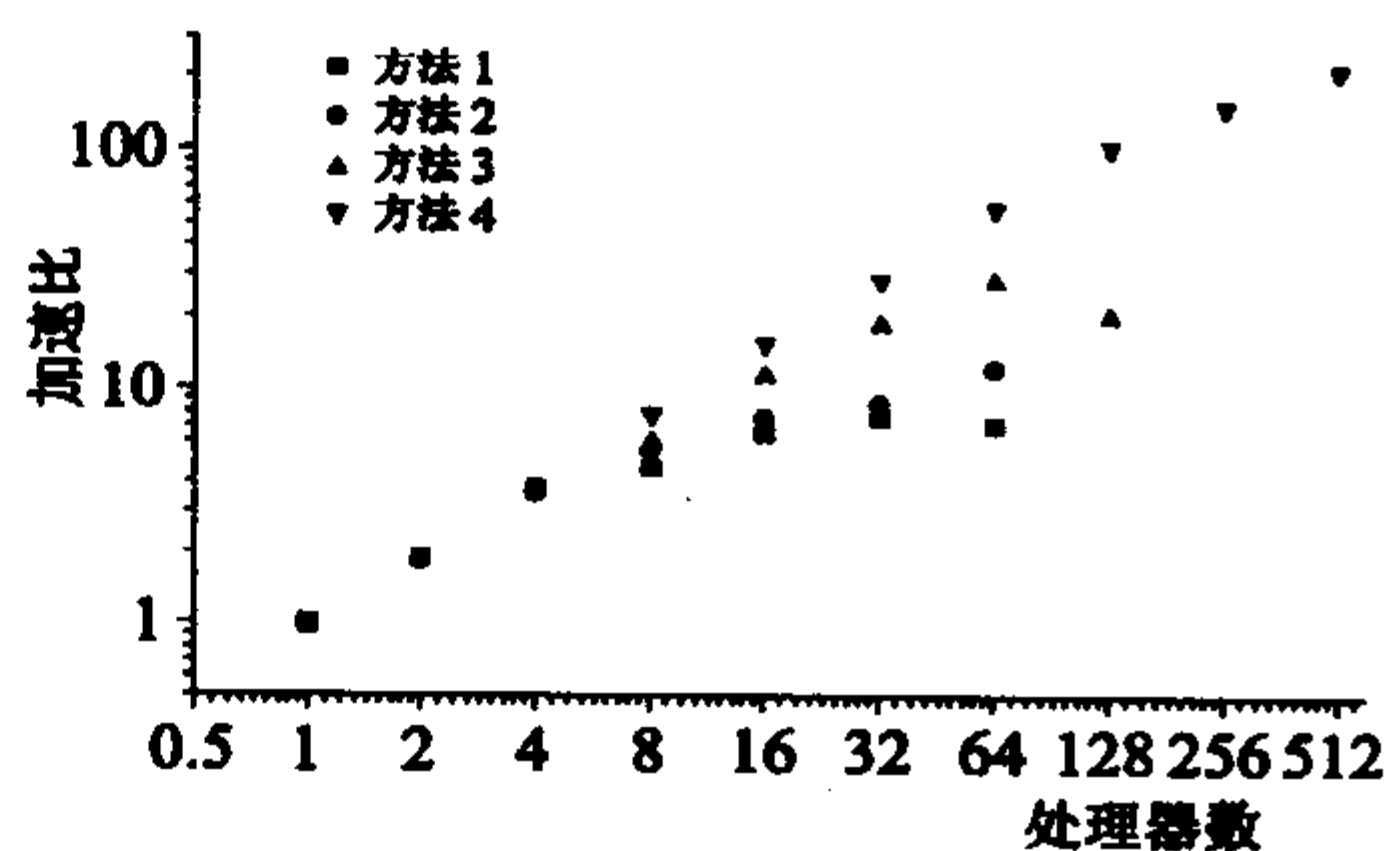
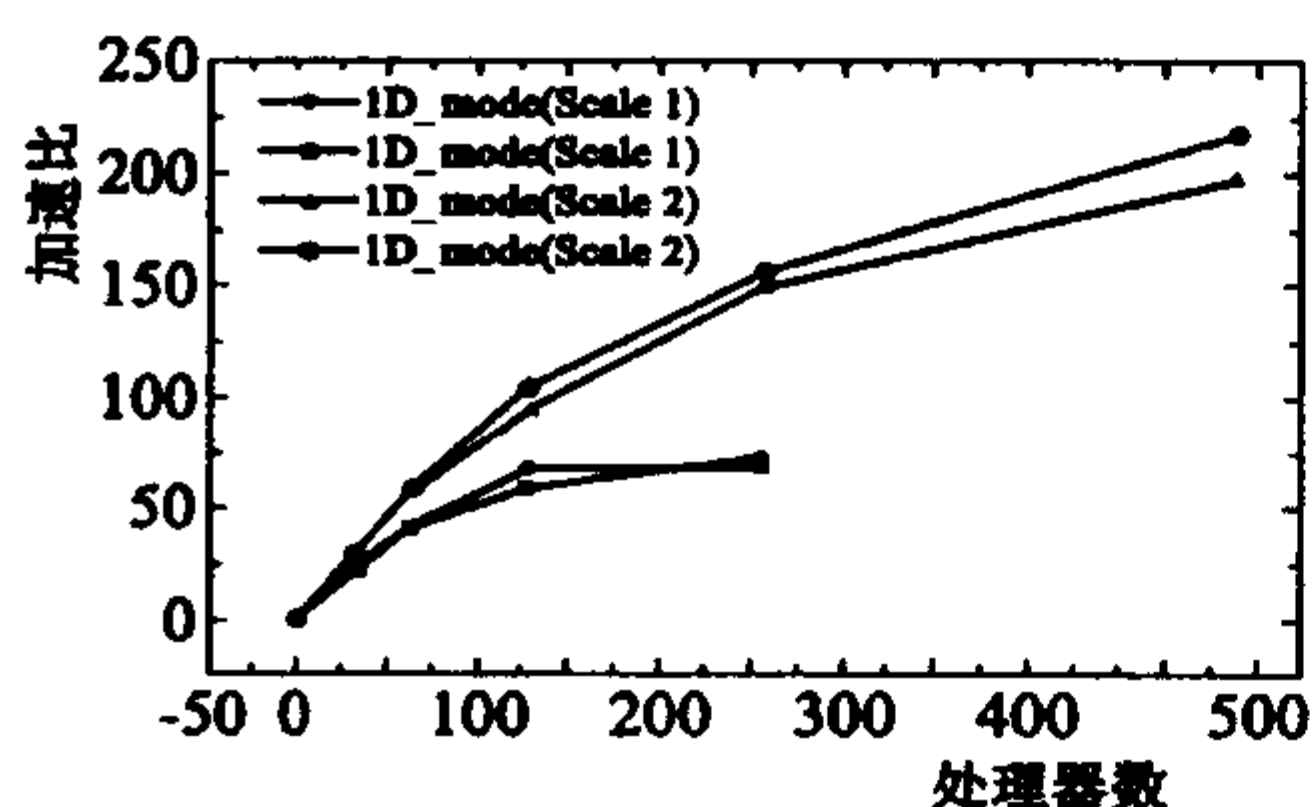


图 7 不同方法的并行加速比

4 总结

本文给出了非定常粒子输运问题 MC 模拟的几种并行 I/O 算法,采用分段随机数发生器,并行计算取得了与串行一致的结果。对大步长问题,采用自适应并行,在 80 个处理器的并行机取得了 56 倍的加速比;对小步长问题,在 512 个处理器计算取得了 200 倍以上的加速比,新方法的可扩展性通过实际问题检验得到了证明。

参考文献

- [1] Deng Li and Xie Zhongsheng. Parallelization of MCNP Monte Carlo Neutron and Photon Transport Code in Parallel Virtual Machine and Message Passing Interface [J]. J. Nucl. Sci. Tech., 1999, 36: p626
- [2] Angela C. Sodan. Applications on a multithreaded architecture: A case study with EARTH-MANNA, Parallel Computing, 2002, 28: 3~33
- [3] P. S. Pacheco. Programming parallel processors using MPI. San Francisco, CA, Morgan Kaufmann, 1995
- [4] M. Snir, S. Otto, S. Huss-lederman, et al. MPI: The complete reference. Cambridge, MA, MIT Press, 1994
- [5] Shyu, C. M., et al. Development of the Monte Carlo library leaset-squares method of analysis for neutron capture prompt gamma-ray analyzes. Nucl. Geophys,

I, 1993, pp241~267

- [6] Deng Li, Liu Jie, Zhang Wenyong et al. The Adaptive Parallel Computation of Time-Dependent Monte Carlo Transport. IEEE Computer Society, 2002 Vol. 2, p489, ISBN 0-7695-1512-6
- [7] Deng Li, Liu Jie, Zhang Wenyong et al. The Parallel Computation of Time-Dependent Monte Carlo Transport. IEEE Computer Society, Proceeding of the 2003 International Conference on Parallel Processing Workshops