1995年7月 第22卷 增刊
西安电子科技大学学报
JOURNAL OF XIDIAN UNIVERSITY
July 1995 Vol. 22 Sup.

江南-Ⅲ机 Monte-Carlo 方法的并行实现*

张春明、杨自强
(中国科学院计算中心 北京 100080)
张正军
(北京航空航天大学 北京 100083)

摘要 以产生(0,1)区间上独立同分布的均匀随机数序列为基础,首先对两种常用的随机数发生器乘同余法和组合发生器的并行实现进行了讨论,对组合发生器的并行实现进行了推荐;其次描述了连续型随机变量抽样的并行实现;最后简述了蒙特卡罗方法在计算数值积分中的应用.

关键词:蒙特卡罗;并行计算;随机数发生器;JN-3;π 运算
中图分类号:O242.1;O211.6

伴随着科技的进步,出现了许多复杂、困难的问题,用通常的解析方法或数值方法难以解决.在这种情况下,一类可行而且是不可缺少的新方法——蒙特卡罗方法(即统计模拟方法)被提出并发展起来.现代电子计算机的出现,又使得在计算机上进行大量模拟试验成为可能.该方法现在已广泛应用于生产、管理系统、运筹学、数学物理、工程技术、仿真技术等科学研究领域[1].特别是并行计算机的出现,为解决大规模统计模拟问题提供了新的环境和条件.江南-Ⅲ多处理机系统正是这种理想的并行机.

1 随机数发生器的并行实现

1.1 乘同余法

自从 J. Von Neumann 在 40 年代提出该方法以来,随机数发生器的研究已取得了许多研究结果.同时随着计算机科学技术的迅速发展,与计算机软、硬件技术相适应的随机数发生器得到了重视和研究.一个好的随机数发生器产生的随机数序列应满足统计独立、均匀、周期长、运算速度快等基本要求.

乘同余法形如: $x_{n+1} \equiv a * x_n \pmod M$, $u_n \equiv x_n / M$

关于乘同余法的并行计算在文献[2]中已有阐述.通过实际计算,发现可能出现下面的

* 收稿日期:1994-12-23

问题.

考虑一个具体的、常用的乘同余随机数发生器：

$$x_{n+1} \equiv 16807 * x_n (\mathrm{mod}\ 2^{31}-1=2147483647)$$

假设使用并行处理器的个数为 ntp，种子为 x_0，以 $ntp=2$ 为例.

令第一个处理器顺序产生 $x_1,x_3,x_5,x_7,\cdots$；第二个处理器顺序产生 $x_2,x_4,x_6,x_8,\cdots$，这样第一、二个处理器分别产生以 x_1,x_2 为种子，以 $A\equiv a^2(\mathrm{mod}\ M)$ 为新的乘子的随机数序列. 由数论理论知其中每个处理器产生随机数序列的周期为

$$\frac{P}{\mathrm{l.c.m.}\ (ntp,P)}$$

P 为以原始 a 为乘子的发生器周期. l. c. m. (ntp,P) 表示 ntp 与 P 的最大公约数. 特别地，当处理器个数 ntp 与 P 互素时，则每个处理器都可获得与原发生器有相同周期的随机数序列. 最后将两个并行处理器产生的随机数按原来顺序连在一起即可.

表面上看，只要事先计算出 A 就可逐步实现下去. 但经进一步分析发现，实际计算并非如此，即使采用双精度，A 的计算也正确无误，中间某一步结果 $A*x_n$ 仍有可能超出计算机所能表示的最大双精度位数(位定义为 bit).

分析如下：现在一般的计算机双精度都采用 53 位精度.

$2^{14}<16807<2^{15}$，$x_n<M=2^{31}-1$，所以 $16807*x_n$ 至多有 46 位，从而在一般的计算机上总是可以实现的. 但 16807^2*x_n 有可能超过 53 位，取模运算后余数结果当然不对. 例如取种子 $x_0=37703$，用一个和二个处理器计算出的前 10 个结果对比见表 1.

表1 用一个和二个处理器计算出的前 10 个结果对比

	$ntp=1$	$ntp=2$		$ntp=1$	$ntp=2$
1	633674321.0	633674321.0	6	898082231.0	1575351078.0
2	792907574.0	792907574.0	7	1552985301.0	796037125.0
3	1261566583.0	1261566582.0	8	507708269.0	749627604.0
4	1043513650.0	1043513660.0	9	1100347552.0	1508274722.0
5	1982454148.0	1699978877.0	10	1559622147.0	2064207076.0

从表 1 中可看出，第三个随机数以后的结果大相径庭. 对 ntp 的其它值也有类似结果. 这说明乘同余法需要满足一定的条件才能并行实现.

1.2 组合同余随机数发生器

近年来，组合同余随机数发生器越来越受到重视. 理论上已经证明它周期长、独立性、均匀性等统计性质都很好. 采用 L'Ecuyer 推荐的组合发生器[3]如下. 该发生器对 16 位的计算机是高质量的发生器. 已通过了大量的统计检验[3]

$$\begin{cases} w_i \equiv 157 * w_{i-1} & (\mathrm{mod}\ 32363) \\ y_i \equiv 146 * y_{i-1} & (\mathrm{mod}\ 31727) \\ z_i \equiv 142 * z_{i-1} & (\mathrm{mod}\ 31657) \\ x_i \equiv (w_i + y_i + z_i - 3) & (\mathrm{mod}\ 32362) \\ U_i = (x_i + 1)/32363 \end{cases} \tag{1}$$

其中种子取 $w_0=y_0=z_0=1$，该发生器的周期为

$$32362 * 31726 * 31656/4 = 8.12543685 \times 10^{12}$$

设 p 为并行处理器个数，令

$$Z_k = \begin{pmatrix} w_{kp+1} & w_{kp+2} & \cdots & w_{kp+p} \\ y_{kp+1} & y_{kp+2} & \cdots & y_{kp+p} \\ z_{kp+1} & z_{kp+2} & \cdots & z_{kp+p} \end{pmatrix}$$

则式(1)的并行格式为

$$Z_k \equiv \begin{pmatrix} a_1^p & & \\ & a_2^p & \\ & & a_3^p \end{pmatrix} \cdot Z_{k-1} \mod \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} \qquad (k = 1,2,\cdots)$$

其中 a_1,a_2,a_3 为 3 个乘子. 组合发生器实质上是把不同的乘同余发生器重新组合起来，以改善随机性，因此其结构极适于并行化. 组合发生器中的每个单独乘同余发生器都较简单，大部分采用整数运算就可实现. 从运算上看，式(1)前 4 步都采用整数运算，最后一步求 U_i 时可根据所研究问题的不同精度要求采用单或双精度运算. 因此速度很快. 若进一步并行化后，就可在较少的时间内产生出所需要的大量高品质随机数. 因此，组合发生器的并行实现值得推荐.

2　连续型随机变量抽样

对随机变量的直接抽样，在得到并行随机数的同时，根据逆变换定理，相应的随机变量的抽样值也就得到了，因此其并行实现极其容易.

对随机变量的舍选抽样，情况复杂，下面给出两种方案：

(1) 舍选抽样效率 E 较高，接近于 1. 如果希望得到 $N=1000$ 个抽样值，则实际试验次数 $KN\approx \frac{N}{E}$，对前 N 次进行并行计算，统计得到抽样值的次数 $N_1(N_1\approx N\cdot E)$，对剩余的 $N-N_1$ 次抽样再进行串行计算.

(2) 更一般的情形是舍选抽样效率 E 不高. 若用第(1)种方法进行并行计算，根据著名的 Amdahl 定律，其加速比有最大可能限制，所以不理想. 这时总试验次数可以用近似试验次数 $K_1=\frac{N}{E}$ 来代替. 随着 N 的增长，近似效果更好. K_1 次抽样可以充分并行抽样得到. 之后统计得到抽样值的次数 KK，KK 与 N 很接近. 若 $KK<N$，不足的可以串行计算；若 $KK>N$，仅仅多出小量抽样，但精度可能更好. 假如希望恰好得到所需的样本，可适当调整 E 的值略大一些. 总之，并行的加速比比方案(1)得到极大提高.

3　数值试验

在江南-Ⅲ并行计算机上，用文中提出的并行 Monte-Carlo 方法模拟概率论历史上著名的 Buffon 随机投针试验[4]计算 π 值. 根据 Monte-Carlo 方法的概率误差估计公式，精度提高一位，试验次数大约增加 100 倍. 这在一般的串行机上是无法忍受的. 采用并行化后就可降低部分试验次数. 由降低方差原理构造的模型用期望值估计

$$\frac{2 \cdot N}{\sum_{i=1}^{N} \sin \pi r_i} \tag{2}$$

近似积分

$$\pi = \frac{2}{\int_0^1 \sin \pi x \cdot \mathrm{d}x} \tag{3}$$

方法一:直接抽样

假设式(2)分母中的 π 值采用精确值.并行计算时间比较见表2.

表2 并行计算时间比较

抽样个数 N	$nlp=1$	$nlp=4$	$nlp=8$	计算结果/s
10^8	1148	293	153	3.1414953069
10^9	11486	2878	1444	3.1415305271

方法二:舍选抽样

为避免式(2)中出现的 π 值,用舍选方案(2)直接并行抽样 $\sin \pi r_i$,并行计算时间比较见表3.

表3 并行计算时间比较

抽样个数 N	$nlp=1$	$nlp=2$	$nlp=4$	$nlp=8$	计算结果/s
10^7	418	218	116	63	3.1408130117
10^8	*3793	2097	1071	542	3.1413065799

注1:带 * 值,因计算时间太长,是估计值.

注2:文中所取抽样次数,用以说明并行 Monte-Carlo 效果.例如:当 $N=10^5$,计算结果为3.141809520;当 $N=8\times10^5$,计算结果为3.1415954867,但并行效果在江南-Ⅲ上不明显,因系统开销比例较大.

由表2、表3容易看出,加速比接近于处理器的个数.当然实际计算 π 时,有更好的四阶、二阶迭代方法[5].在江南-Ⅲ上已成功地并行计算 π 至五百万位.文中计算此例的目的是说明并行 Monto-Carlo 的应用.

参 考 文 献

1 Bratley P, Fox B L, Schrage L E. A Guide to Simulation. Second Edition. Springer Verlag: World Publishing Corporation, 1987

2 王嘉谟,沈 毅.并行计算方法:下.北京:国防工业出版社,1992

3 L'Ecuyer P. A Portable Random Number Generator for 16-Bit Computers. In: Hogan P. Modeling and Simulation on Microcomputers. San Diego, California: The Society for Computer Simulation, 1987. 45～49

4 中国科学院计算中心概率统计组.概率统计计算.北京:科学出版社,1979

5 Balley D H. The Computation of π to 29,360,000 Decimal Digits using Borweins' Quartically Convergent Algorithm. Mathematics of Computation, 1988, 50(161): 283～296

(编辑:郭 华)

Monte-Carlo method and its application implemented on JN-3 parallel computer

Zhang Chunming　Yang Ziqiang
(Computing Center, Academia Sinica, Beijing, 100080)
Zhang Zheng jun
(Beijing University of Aeronautics and Astronautics, Beijing, 100083)

Abstract

Since the generation of pseudo-random number sequence distributed over the interval (0, 1) is the basis of simulation, this paper first discusses the parallel implementation of two commonly used pseudo-random number generators: the multiplicative congruential and the combinative generator. The parallel implementation of a combinative generator is recommended. Then, the parallel implementation of sampling the continuous random variables is described. Finally, an application of the parallel Monte-Carlo method in numerical integration is illustrated, which shows a high speedup.

Key Words: Monte-Carlo; parallel computation; random number generator; JN-3; π computation

(上接 20 页)

fuzziness of this kind of sample. We also study the relationship between a small sample and an incomplete knowledge sample. We introduce the principle of information diffusion and the normal diffusion method to improve the recognition procision when the sample is incomplete.

Key Words: knowledge sample; incomplete sample; small sample; fuzzy information; information diffusion; normal information diffusion