高性能科学计算的特征分析及其实用方法研究*)

王文义[1] 王若雨[2] 董绍静[3]

(中原工学院并行处理技术研究所 郑州 450007)[1] (河南电力职工大学网络中心 郑州 450051)[2]

(肯塔基大学超级计算中心 美国列克星敦 40506)[3]

摘 要 任何高性能科学计算(HPC)课题都是一项复杂的系统工程,其具体的应用效率要受到来自硬件和软件等许多因素,主要如并行算法、流水线技术、层次存储器技术和网络互联结构等的制约,诸因素既互相独立又互相关联。本文从一个典型的高性能科学计算——格点量子色动力学研究模型入手,在分析了 HPC 所涉硬软件存在的一些共性特征的基础上,总结出一些能够改善高性能科学计算应用效率的方法。通过对这些方法长期的应用实践和专门实验,证明它们是有效的。

关键词 高性能科学计算,格点量子色动力学,线性模型,费米矩阵,层次存储器技术

Study of Analysis of the Features Based on High Performance Science Computing and its Practical Methods

WANG Wen-yi[1] WANG Ruo-yu[2] DONG Shao-jing[3]

(Institute of Parallel Processing Technology, Zhongyuan Institute of Technology, Zhengzhou 450007, China)[1]

(Network Center, Henan University of Electric Power and Workers, Zhengzhou 450051, China)[2]

(Supercomputing Center, University Kentucky, Lexington 40506, USA)[3]

Abstract Any one of HPC's projects is a complicated systematic engineering and its efficiency in practical application is restricted by many factors from hardwares and softwares such as parallel algorithm, pipelining technology, hierarchical memory technology and network connection structure which are both independent of and interrelated to each other. This paper, by means of studying the typical HPC—Lattice Quantum ChromoDynamic(LQCD) model and analyzing the common features of the hardwares and softwares involved in the HPC, works out some methods which can enhance the application efficiency of HPC. The long-period application and special experiments prove that these methods are effective.

Keywords High performance science computing, Lattice quantum chromodynamic, Linear model, Fermion matrix, Hierarchical memory technology

1 引言

高性能科学计算,由于它在理论科学和实验科学之间建立起了相辅相成、彼此印证的关系,现在已是人类科学研究必不可少的方法之一,其发展水平也成为衡量一个国家综合国力的主要指标。

高性能科学计算是一项系统工程,它受到许多复杂因素的制约,即使有了 3T 性能的计算机,也并不意味着问题就能圆满解决。所涉因素,如多功能部件、流水线技术、存储器分层技术、网络互连结构、并行算法和并行程序设计等等[1,2,10],只有在各个方面做到精心合理的配合,才会有助于目标问题的解决。我们在详细研究典型的高性能科学计算——格点量子色动力学(Lattice Quantum ChromoDynamic, LQCD)问题的基础上,专门对高性能科学计算和并行计算机系统的一些共性特征进行了深入分析,并由此提出一些看法和体会,以飨读者。

2 大规模线性计算模型的基本特征

属于场论的 LQCD 研究[3-5],在 HPCC 计划中被列为 21 世纪的"重大挑战性"课题之一,完成课题测量所必需的 200 个组态对计算机计算性能的需求大约为 54Teraflop·年。

2.1 LQCD 研究实例

根据 LQCD 理论,夸克-夸克和核子-核子两个相关函数可分别表示为

$$\langle q_j \cdot \bar{q}_i \rangle = \langle M_{j,i}^{-1} \rangle$$

和

$$\sum_x e^{-i\vec{p}\cdot\vec{x}} \langle N^{\alpha}(x)\bar{N}^{\alpha'}(0)\rangle$$
$$= \varepsilon^{abc}\varepsilon^{a'b'c'} \sum_x e^{-i\vec{p}\cdot\vec{x}} \langle 0|[M_{aa}^{-1}{}^{\alpha\alpha'}(x,0)$$
$$\underline{M_{\sigma\zeta}^{-1}{}^{bb'}(x,0)M_{\beta\beta}^{-1}{}^{\alpha'}(x,0)} + M_{a\varepsilon}^{-1}{}^{\alpha\alpha'}(x,0)$$
$$\underline{M_{\varepsilon\xi}^{-1}{}^{bb'}(x,0)M_{\xi a}^{-1}{}^{\alpha'}(x,0)}]|0\rangle$$

其中,$M=[(\gamma_2\gamma_5)M(\gamma_2\gamma_5)^{-1}]^T$,即著名的 Fermion(费米)矩阵。所有的 γ 为 4×4 复数矩阵,而对于核子-轴向电流-核子相关函数,由于连接嵌入和非连接嵌入两项的加入,因此将更为复杂。前者是由各种夸克传播函数形成的,后者则可用下式表示(夸克环):

$$\sum_x \{Tr[\gamma_\mu\gamma_5 U_\mu(x)^{gh}M^{-1}{}^{hg}(x+\hat{\mu},x)] +$$
$$Tr[\gamma_\mu\gamma_5 U_\mu^{+}(x)^{gh}M^{-1}{}^{hg}(x,x+\hat{\mu})]\}$$

*)受国家自然科学基金"基于自适应搜索的快速运动估计算法研究"(编号 60075006),河南省研发专项资金项目"提高高性能计算机有效速度研究"(编号 0641060401)资助。**王文义** 教授,硕士生导师,研究方向为高性能科学计算;**王若雨** 硕士研究生;**董绍静** 旅美博士,教授,主要研究领域为计算物理学与并行算法。

至此，如何获取 Fermion 矩阵 $M_{i,j}$ 的直列区，即如何求解夸克传播函数以及对角和非对角区这两种必需的数据就成为关键。

2.2 大规模 Fermion 矩阵 M

根据 K. G. Wilson[5] 提出的求解途径，可以把 LQCD 划分函数 Z 用近似的函数积分形式表示为

$$Z=\int\prod_i \mathrm{d}\bar{q}_i\mathrm{d}q_i\mathrm{d}U_i\mathrm{e}^{-s}$$

且

$$S=\int\mathrm{d}^4x\{\frac{1}{4}F_{\mu\nu}F^{\mu\nu}+O(a^2)\}+\bar{q}(x)\cdot M\cdot q(y)$$

其中 S 为格点(Lattice)作用力，M 为 Fermion 矩阵，$q(y)$ 和 $\bar{q}(y)$ 为 Grassman 变量。为了方便计算机处理，需要首先对它们求积分，以得到行列式 $\det(M)$，然后再把 Z 变换成

$$Z=\int\prod_i \mathrm{d}U_i\ \det(M)\mathrm{e}^{-S_g}$$

继而得到最终计算模型：

$$F(U_i)=\det(M(U_i))\mathrm{e}^{-s_g(U_i)}$$

$$\langle O\rangle=\frac{1}{z}\int\mathrm{d}U_1\int\mathrm{d}U_2\cdots\int\mathrm{d}U_N$$

$$O(M(U_1,\cdots,U_N))\cdot F(U_1,\cdots,U_N)$$

但是，要想通过对数百万个变量 U 进行积分，以求得 Z 的精确解是不切实际的。可以通过 Monte Carlo 方法对组态

$$\langle\dot{O}\rangle=\frac{1}{Z}\int\prod_i \mathrm{d}U_i\,\dot{O}(U)\det(M)\mathrm{e}^{-S_g(U)}\approx\frac{1}{N}\sum_{i=1}^{N}\dot{O}(U_i)$$

求平均值来达到计算 $\dot{O}$ 的目的。LQCD 理论把 Fermion 矩阵 M 看作是一个 Non-hermitian (非厄密)复数矩阵：

$$M_{i,j}=M_{aax,bby}=\delta_{a,b}\delta_{\alpha,\beta}\delta_{x,y}-k\sum_{\mu=1}^{4}[(1-\gamma_\mu)_{\alpha}^{\beta}$$

$$U_\mu(x)_{ab}\delta_{x,y-\hat{\mu}}+(1+\gamma_\mu)_{\alpha\delta}U_\mu^{+}(x)_{ab}\delta_{x,y+\hat{\mu}}]$$

其中，x,y 是四维坐标，所有 γ_μ 都是 4×4 复数自旋矩阵，μ 被用于选择边界条件。对 Fermion 矩阵 M，在众多格点规模中，即使取较小的规模 $16^3\times24$，其容量也有 2,359,596；而对于非常有价值的 $24^3\times32$，其容量将达到 10,616,832。M 是一个超大规模稀疏矩阵，其非零元素有 500M 个以上，仅此一项就要占去计算机 4GB 的存储空间！所以，对计算机的时空资源都提出了相当苛刻的要求。

2.3 高性能科学计算模型的共性特征

在求解 LQCD 的过程中，需要不断地对以 Fermion 矩阵 M 为首的大型矩阵进行行列式、矩阵求逆、旋转等运算，并且几乎所有的关键计算部分都要受到矩阵乘[8]特性的支配，这部分处理要耗去绝大部分计算时间。

鉴于上述现象，我们曾在国内外做过大量调查，对象包括石油勘探与地震资料处理、理论物理研究、材料设计和气象气候研究等不同行业和不同类型的高性能计算课题。调查结果发现，在几乎所有被调查的对象中，除了有少量特例之外，大部分课题的核心处理过程也都与 LQCD 研究相似，求解过程最终都归结到了一处，即必须花费大量时间对一些“超级”矩阵进行大规模线性运算才能得到结果。由此我们得到的启发是——如何找出大规模线性运算对象与高性能计算机的运算部件间的对应与配合关系，将是提高解决同类问题实际应用效率的一个重要途径。

3 基本线性计算模型与层次存储器技术

3.1 基本线性计算模型

假设运算对象分别定义为 $A(n,n)$，$B(n,n)$，$C(n,n)$，$D(n)$，$E(n)$，$F(n)$ 和常量 a，则通常基于计算机语言表示形式的线性计算包括：

(1)向量或矩阵赋值

① 向量赋值：$D_i=E$

② 矩阵赋值：$A_i=B$

(2)向量与向量运算：$D_i=E+a*F$

(3)矩阵与向量乘：$E_i=A*D$

(4)矩阵与矩阵乘：$C_i=A*B$

3.2 层次存储器技术

高性能计算机，无论是工作站、超级巨型计算机还是远程的虚拟集群系统，其存储器设计通常都采用了分层技术[1,2]，少的二三层，多的五六层甚至七八层等。若以六层表述，则可表示为：

0 层：寄存器

1 层：cache

2 层：主存储器

3 层：局部共享存储器

4 层：全局共享存储器

5 层：硬盘

层次存储器的特点遵从层与层之间容量递增而速度递减的规律。随着微电子技术的飞速发展，高性能计算机各层存储器的容量能够做得越来越大，如新一代的 HP/PA-RISC 8900 处理器，最大可以支持到 256GB DDR 内存，处理器内部配置的一、二级缓存分别达到了 1.5MB 和 64MB 。

尽管 cache 的容量也在不断增大，但较之高性能科学计算的大规模数据来说毕竟还是非常有限的，不可能一下子把全部数据都放入 cache 参加运算，尚需要借助算法的配合。目前，基于“块算法”的技术在国内外已被广泛应用，即需要先对目标数据进行分解，使之每个块数据的大小尽量与所选用的存储器容量相等或相近。现在大多数高性能计算机支持的编译器也都增加了可以在程序中由人工指定存储器层次(效果见表 2)的功能。因此，如果我们能够把分解后的数据都定位到速度较高的 cache 上，从而尽可能地缩短 cache 延迟时间，无疑会明显提高计算的效率。

4 cache 命中率与线性模型的关系

4.1 理论峰值速度与应用速度

高性能计算机的理论峰值速度与用户应用速度之间普遍存在着较大差异(如表 1 所示，表中的 Pentum 与 DEC21146-AA 属远程虚拟集群系统)，现在基本上已经成为共识。同时，我们在研究 LQCD 等高性能科学计算的实践过程中也发现，在计算机的三大延迟中，cache 延迟因素对应用速度的影响要远远大于通信延迟和译码延迟。

表 1 使用 8 个处理器的 LQCD Monte Carlo 代码测试结果（速度单位：Mflops)

机 型	峰值速度/处理器	应用速度/处理器	有效速度/峰值速度(%)
Pentum	75	1.17	1.0
DEC21146-AA	266	40.8	15.3
T3E1200	1200	584.4	48.7
Origin2000	390	170.9	43.8
HP/Exemplar2000	400	172.8	43.2

4.2 线性计算模型的 cache 命中率

4.2.1 cache 命中率

我们可以把 cache 命中率 CHR(Cache Hitting Ratios)理解为：

CHR = 指令执行总次数/访问数据的总个数，即 cache

中平均每个数据参加运算的次数。该平均值越大，则表示cache的命中率越高。

在忽略了相对较小的通信延迟与译码延迟因素之后，应用速度 V_A、峰值速度 V_P、cache 缺失率 CMR(Cache Missing Ratios)和 CHR 间的关系可以近似表示为

$$V_A \approx V_P * (1-CMR) = V_P * CHR$$

4.2.2 线性计算模型的 CHR

根据 3.1 的定义，若我们把程序执行中的“赋值”和“存数”操作也分别看作运算和访问行为，则有

①量赋值 $D_i := E$

共有 n 次赋值和 $2n$ 次访问，即

$CHR = n/(2n) = 1/2$

②阵赋值 A:= B

共有 n^2 次赋值和 $2n^2$ 次访问，即

$CHR = n^2/(2n^2) = 1/2$

③向量与向量运算 $D_i := E + a * F$

共有 $3n$ 次乘法运算、加法运算、赋值运算和对 $3n$ 个数据的访问，所以

$CHR = 3n/(3n) = 1$

④矩阵与向量乘 $E_i := A * D$

每个 E 元素，需进行 $2n$ 次乘法、加法和赋值运算，n 个 E 元素的运算总次数应为 $2n^2$；另外，需要对 n^2 个 A 元素、n 个 D 元素的访问和对 n 个 E 元素的存入操作，总次数应为 n^2+2n。所以

$CHR = 2n^2/(n^2+2n)$

当 $n \to \infty$ 时，$CHR \approx 2$。

⑤矩阵与矩阵乘 $C_i := A * B$

每个 C 元素需进行 n 次乘法、$n-1$ 次加法和 1 次赋值运算，n^2 个 C 元素的运算总次数应为 $2n^3$；另外，需要对 A，B 两个矩阵共 $2n^2$ 个数据的访问和对 n^2 个 C 矩阵数据的存入操作，总次数应为 $3n^2$。所以

$CHR = 2n^3/(3n^2) = (2/3) * n$

上述分析结果表明，在 5 种线性模型中，其 cache 命中率 CHR 是逐渐增高的，直到矩阵相乘时出现 CHR 与代表矩阵规模的参数 n 成正比的情况为止。当 $n \to \infty$ 时，CHR 也达到了无穷大，它反映出当矩阵数据量增大时，数据被命中的概率也将随之增加。

5 实验测试

表 2 在 LQCD 中设定较高速存储器前后的 cache 延迟时间(秒)

线性计算类型	默认存储器时的 cache 延迟时间	设定高速存储器时的 cache 延迟时间	时间提升率
向量赋值 $D_i := E$	298.2	156.5	47.5(%)
矩阵与向量乘 $E_i := A * D$	348.2	273.1	21.6(%)
A 矩阵求逆	5142.6	3994.9	34.0(%)

在 HP/SPP2200 (64 个处理器，单处理器的峰值速度为 800Mflops)机器上使用 8 个处理器，分别用 LQCD 中矩阵乘等 3 种不同线性计算模型代码和 HP 公司的通用程序库代码进行测试，测试数据如表 2 和表 3 所示(注：所用程序为 FORTRAN 90 程序[6,7,9]，数据规模为矩阵 3000 * 3000、向量 3000、数据类型 real * 8，此时单处理器峰值速度为 400Mflops)。

表 3 峰值速度为 400Mflops 时不同线性计算模型的 CHR 与 V_A

计算类型	CHR	实际应用速度 V_A/处理器
$D_i := E\ (A_i := B)$	20.2%	80.8Mflops
$D_i := E + a * F$	27.7%	110.8Mflops
$E_i := A * D$	55.3%	221.2Mflops
$C_i := A * B$	78.1%	312.4Mflops

从表 2,3 中的数据不难看出，如果能够在程序设计中由人工把数据分解之后的数据指定在较高速度的存储器中，是可以明显提高计算机的实际应用速度的。同时，在主要由不同线性计算类型组成的大规模高性能计算中，由于彼此所获得的计算机应用速度存在很大差异，因此建议应尽量使用 cache 命中率较高的线性计算模型，而对 CHR 较低的模型则应少用或改用其它方法予以解决。

结束语 在高性能科学计算中，用户常常要受到计算资源的限制，所以他们最关心的莫过于是能够尽快地得到计算结果，而最无奈的则是长时间的等待过程，因此如何缩短计算时间，这是值得很多从事高性能科学计算的用户去研究的问题。作者通过对高性能计算机以及对高性能计算课题的长期实践研究，针对其中存在的大规模线性计算模型、存储器分层技术、cache 命中率等一些共性特征进行了分析、归类和总结，提出了一些在应用中值得考虑的方法，但愿对关心该领域的读者能有些启发与帮助。

参 考 文 献

[1] Hennessy J L, Patterson D A. Computer Architecture: A Quantitative Approach. Third Edition (英文版). 北京：机械工业出版社，2004：390-443，814-834

[2] Hwang K. Advanced Computer Architecture: Parallelism Scalability Programmability. New York：McGraw-Hill Inc，1993：188-194

[3] Dong Shaojing, Wang Wenyi. A parallelization test on SPP1200 of a large scalar matrix computation——Lattice Quantium Chromodynamic(QCD) and the hierarchi-cal problem// Proceeding of Convex/HP User Group world Wide Conference. Dallas, March 1996

[4] Mathur N, Dong Shaojing. Study of stochastic estimates of Quark loops with unbiased subtraction. Lattice 2002 at Boston, Nucl. Phys. B(Proc. Suppl.)，2003，119

[5] Wilson K G. Phys. Rev，1974，D10：2445

[6] Braimerd W S, Golberg C H, Adams J C. Programmer's Guide to Fortran 90. New York，McGraw-Hill，1990

[7] Gropp W, Lusk E, Skjellum A. Using MPI: Portable Parallel Programming with the Message Passing Interface. 2nd Edition. Cambridge，MA：MIT Press，1999

[8] Bertsen J. Communication-Efficent Matrix Mutip- lication on Hypercubes. Parallel Computing，1990：335-342

[9] MPICH. http://www.mcs.anl.gov/mpi/mpich

[10] 陈国良，等. 并行算法实践. 北京：高等教育出版社，2004