

PC 集群系统中 MPI 并行矩量法研究

张 玉 王 萌 梁昌洪 谢拥军

(西安电子科技大学电子工程学院 西安 710071)

摘 要: 针对复杂环境的电磁兼容分析中计算量过大、耗时太长的障碍, 该文组建了一个高性能 PC 集群系统, 以此为硬件平台研究了 MPI 并行环境下的并行矩量法。论文首先给出了与并行共轭梯度法求解矩阵方程对应的矩量法矩阵的棋盘块划分方式, 然后详细地讨论了并行共轭梯度算法求解矩量法矩阵方程的并行实现。作为应用实例计算了某飞行器模型的散射特性, 并测试了在 PC 集群系统中本文并行矩量法程序的性能。

关键词: 电磁散射, PC 群集系统, 并行计算, 矩量法

中图分类号: 0441 **文献标识码:** A **文章编号:** 1009-5896(2005)04-0647-04

Study of Parallel MoM on PC Clusters

Zhang Yu Wang Meng Liang Chang-hong Xie Yong-jun

(School of electronic engineering, Xidian Univ., Xi'an 710071, China)

Abstract Prohibitive computation resources and too much time are needed for EMC analysis of complicated EM environment. To overcome this drawback, a parallel algorithm that combined MoM with MPI functions is studied in this paper. The tessellation scheme is employed for parallel filling the impedance matrix and parallel conjugate gradient method is used to solve the matrix equation. The performance of the parallel code on PC clusters is analyzed and numerical results show its efficiency.

Key words Electromagnetic scattering, PC clusters, Parallel computation, MoM

1 引言

随着计算机技术的发展, 矩量法得到了越来越广泛的应用^[1], 但是, 由于受到计算机资源的限制, 对电大尺寸目标的电磁散射特性的分析, 一直以来难于直接在 PC 机上实现。国内外的研究者对并行矩量法的研究工作主要依赖于大型计算机或工作站^[2], 如 IBM(RS6000), CRAY(Digital alpha), Meiko(Sun), SGI(MIPS)和 Convex(HP)等, 这些工作平台价格极其昂贵。最近, 以 PC 机网络群集系统为工作平台的大型计算系统蓬勃发展, 事实上, 由台式计算机构成这样的系统无需任何额外的投资, 这为低成本的高性能计算提供了硬件基础。

为了实现电大目标电磁特性分析, 同时节省硬件成本和计算时间, 笔者曾研究了 MPI 支持下的 PC 机组成的网络群集系统下的并行矩量法^[3], 用于解决矩量法由于计算量和内存需求量大而导致单机目前无法计算的问题。作者在文献[3]

中研究的是并行高斯消元法求解线性方程组及其对应的矩量法矩阵并行填充方法。对于实际的电大目标电磁特性矩量法分析来说, 并行高斯消去法由于运算量大而不是一种广泛采用的方法, 人们往往采用迭代方法(CG)求解方程, 为此, 本文将进一步深入研究并行方法求解并行 RWG 基函数矩量法^[1]矩阵方程以及相应的矩阵填充方式, 并将实例考察其运行于 PC 集群中的性能。

2 并行矩阵填充与并行 CGN 方法

并行矩量法就是运用并行技术填充矩量法的阻抗矩阵和求解相应的矩阵方程, 因为矩量法问题的基础是矩阵, 即一些数(或代表数的变量)的二维数组。一个 $M \times N$ 的矩阵有 M 行和 N 列元素, 对于顺序编程来说, 这种结构可以用二维数组来加以表示。对于其并行填充方式, 下面将结合 CGN (Conjugate Gradient Norm)算法求解矩阵方程详加讨论, 为此, 首先给出本文采用的 CGN 迭代算法。

2.1 CGN算法

CGN 方法求解复线性方程组的一般步骤为:

```

guess  $X_0$ 
 $r_0 = AX_0 - b$ 
 $P_1 = -A^+ r_0$ 
iterate( $n=1,2,\dots$ )

 $\alpha_n = -\frac{\langle AP_n, r_{n-1} \rangle}{\|AP_n\|^2} = \frac{\|A^+ r_{n-1}\|^2}{\|AP_n\|^2}$ 

 $X_n = X_{n-1} + \alpha_n AP_n$ 

 $\beta_n = \frac{\|A^+ r_n\|^2}{\|A^+ r_{n-1}\|^2}$ 

 $P_{n+1} = -A^+ r_n + \beta_n P_n$ 
    
```

2.2 棋盘状划分的并行矩阵填充方式

由前可以看出 CGN 方法不仅需要计算阻抗矩阵 A 与未知量 x 的矩阵向量乘运算, 而且还要做阻抗矩阵的共轭矩阵相关的矩阵向量乘法运算。因此, 在并行计算时, 为了使进程比较有效地同时计算矩阵向量乘与共轭矩阵向量乘运算, 本章将整体矩阵采用图 1 所示的棋盘状分布方式分配到各个进程, 同时也给出了阻抗矩阵的共轭矩阵的进程分布方式。

对于量量法的矩阵问题, 显然, 只要 MPI 二维拓扑^[4]即可实现物理模型与逻辑模型的对应关系, 如图 1 所示。比如将一个矩阵分解成 9 个不同的子区域, 然后将这 9 个区域对应到不同的进程上, 对应关系可以由图 1(b) 表示。这样, 通过 Cartes 坐标系, 可非常方便地知道总体矩阵元素在各个进程中的位置。

这种棋盘状分布的 RWG 基函数量量法矩阵地并行填充可以用伪代码概括如下:

```

Do  $q = 1, \text{NumTriangles}$  → loop over Triangles
Do  $q = 1, \text{NumTriangles}$ 
  compute_edge_interactions( $q, p$ )
Do  $e = 1, 3$  → loop over edges
   $m = \text{edge\_num}(q, e)$  → reference from triangle to edge
  If ( $m.$  NE.-1. and  $.m > \text{coords}(1) * n \times 2$ .
  and  $.m \leq (\text{coords}(1) + 1) * n \times 2$ ) then
     $m = m - \text{coords}(1) * n \times 2$  → adjust for local numbering
    in this processor's sub-block
  Do  $f = 1, 3$  → loop over edges
     $n = \text{edge\_num}(p, f)$  → reference from triangle to edge
    If ( $n.$  NE.-1. and  $.n > \text{coords}(2) * n \times 2$ .
    and  $.n \leq (\text{coords}(2) + 1) * n \times 2$ ) then
       $n = n - \text{coords}(2) * n \times 2$  → adjust for local numbering
      in this processor's sub-block
     $A(m, n) = A(m, n) + \text{delta}$ 
  Endif
Enddo
Endif
Enddo
Enddo
Enddo
    
```

其中, $nx2$ 是分块方阵的维数, $\text{coords}(1), \text{coords}(2)$ 是进程的虚拟拓扑坐标。

2.3 并行CGN方法求解矩阵方程

接着我们讨论与这种填充方式相对应的并行 CGN 求解算法。由前可知, CGN 迭代解法中的关键计算是矩阵向量乘法运算, 这种运算在串行程序中是不难实现的。对于并行化的 CGN 算法, 问题的难点在于几个矩阵向量乘, 为简单起见, 归结为 Ax, A^+x , 只要把 Ax, A^+x 求出来, 问题就解决了。因此首先讨论如何用并行求解 Ax, A^+x 。对于 Ax , 以矩阵分成 3 行 3 列为例, 如图 2。

图 2 中, a 下标的第 1 位表示进程, x 下标的前 3 位表示向量段存在于该数字表示的 3 个进程中。

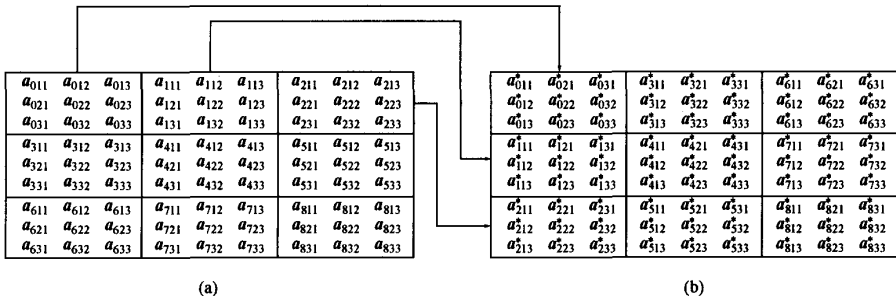


图 1 矩阵 A 与矩阵 A^+ 的进程分布
(a 的下标第一个字母表示所在进程)

a_{011}	a_{012}	a_{013}	a_{111}	a_{112}	a_{113}	a_{211}	a_{212}	a_{213}	x_{0361}
a_{021}	a_{022}	a_{023}	a_{121}	a_{122}	a_{123}	a_{221}	a_{222}	a_{223}	x_{0362}
a_{031}	a_{032}	a_{033}	a_{131}	a_{132}	a_{133}	a_{231}	a_{232}	a_{233}	x_{0363}
a_{311}	a_{312}	a_{313}	a_{411}	a_{412}	a_{413}	a_{511}	a_{512}	a_{513}	x_{1471}
a_{321}	a_{322}	a_{323}	a_{421}	a_{422}	a_{423}	a_{521}	a_{522}	a_{523}	x_{1472}
a_{331}	a_{332}	a_{333}	a_{431}	a_{432}	a_{433}	a_{531}	a_{532}	a_{533}	x_{1473}
a_{611}	a_{612}	a_{613}	a_{711}	a_{712}	a_{713}	a_{811}	a_{812}	a_{813}	x_{2581}
a_{621}	a_{622}	a_{623}	a_{721}	a_{722}	a_{723}	a_{821}	a_{822}	a_{823}	x_{2582}
a_{631}	a_{632}	a_{633}	a_{731}	a_{732}	a_{733}	a_{831}	a_{832}	a_{833}	x_{2583}

图2 矩阵与向量相乘

若令 $Ax = b$, 则其分块矩阵运算可以表示为:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

相应的进程为

$$\begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix} \begin{pmatrix} 0,3,6 \\ 1,4,7 \\ 2,5,8 \end{pmatrix} = \begin{pmatrix} 0,1,2 \\ 3,4,5 \\ 6,7,8 \end{pmatrix}$$

矩阵 A 分为 9 个分块矩阵, 每块矩阵分别存在一个进程中, 向量 x 分为 x_1, x_2, x_3 3 个向量, x_1 存在进程 0, 3, 6 中, x_2 存在进程 1, 4, 7 中, x_3 存在进程 2, 5, 8 中。 A_{11}, A_{12}, A_{13} 分别与 x_1, x_2, x_3 相乘, 则进程 0, 1, 2 分别有向量 b_1 , 3 个不同进程的 b_1 相加后即可得到整体的 b_1 , 结果存在 0 进程。同理, 进程 3 得到整体的 b_2 , 进程 6 得到整体的 b_3 , 这一通信过程可以用图 3 表示, 其中 x, b 下括号内数字表示要利用该数字表示的进程中的向量段。把 b_2, b_3 传递给 0 进程即可得到所求的向量 b 。

类似地, 对于 A^+x , 以矩阵分成 3 行 3 列为例。则

$$\begin{bmatrix} A^+_{11} & A^+_{12} & A^+_{13} \\ A^+_{21} & A^+_{22} & A^+_{23} \\ A^+_{31} & A^+_{32} & A^+_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

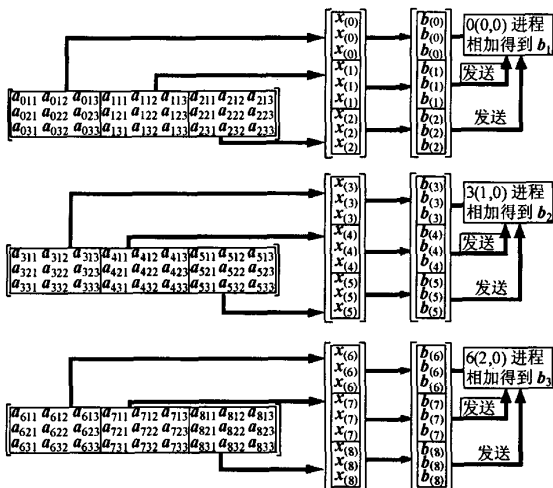


图3 各进程的子矩阵与向量相乘

相应的进程为

$$\begin{pmatrix} 0 & 3 & 6 \\ 1 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix} \begin{pmatrix} 0,1,2 \\ 3,4,5 \\ 6,7,8 \end{pmatrix} = \begin{pmatrix} 0,3,6 \\ 1,4,7 \\ 2,5,8 \end{pmatrix}$$

矩阵 A^+ 分为 9 个分块矩阵, 每块矩阵分别存在一个进程中, 向量 x 分为 x_1, x_2, x_3 3 个向量, x_1 存在进程 0, 1, 2 中, x_2 存在进程 3, 4, 5 中, x_3 存在进程 6, 7, 8 中。 $A^+_{11}, A^+_{21}, A^+_{31}$ 分别与 x_1, x_2, x_3 相乘, 则进程 0, 3, 6 分别有向量 b_1 , 3 个不同进程的 b_1 相加后即可得到整体的 b_1 , 结果存在 0 进程。同理, 进程 1 得到整体的 b_2 , 进程 2 得到整体的 b_3 , 把 b_2, b_3 传递给 0 进程即可得到向量 b 。

为了程序实现上的简单, 0 进程获得了矩阵向量相乘的结果 c 向量以后, 广播给所有进程。这样就可以方便地进行 CG 算法要求的其他矩阵向量乘、向量加、向量减等等运算。

3 MPI 工作网络

在计算资源成本有限的情况下, 本文应用的 PC 集群系统, 其模型如图 4 所示, 系统主要参数设置如下: 10 节点; 每个节点 2.8 GHz CPU, 1.0 GByte 内存; 100/1000M 高速网卡; 1000 MByte 交换机。

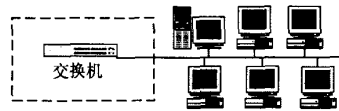


图4 MPI 工作网络结构图

4 性能评价与应用

假设某飞行器表面按照最大边不超过 1/10 波长来划分三角形 (局部需要细化三角形), 那么 100MHz 时其散射特性的 MoM 计算 (采用双精度) 具有几千个未知量, 目前这在普通 PC 机上难以实现的。为了测试算例, 本文选取了 12150 个未知量的剖分情况, 将并行矩量法运行于上述 PC 集群系统, 则可以方便地得到其散射特性。假设 Ez 沿着 -x 方向入射, 则表面电流 (分贝值) 分布和 RCS 结果如图 5 所示。

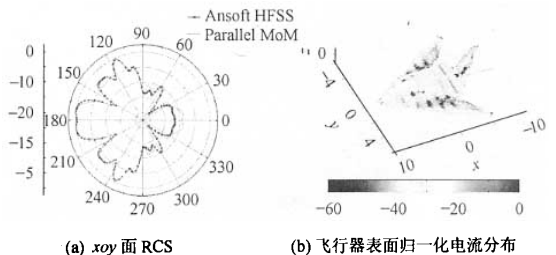


图5 并行矩量法计算结果

利用不同 CPU 个数时, 矩量法矩阵填充与求解测试时间(收敛精度定为 0.05) 的比较情况列在表 1 中。当 CPU 个数从 4 变到 9 时, 矩阵填充加速了 1.546 倍, 方程求解加速了 2.25 倍。可见, 由于棋盘状划分的矩阵填充时不同节点有重复运算, 所以不能达到期望的加速 2.25 倍, 而求解时各节点之间无重复运算, 所以可以加速 2.25 倍。总体加速比例仍然可以高达 2.203 倍, 略小于期望值 2.25。可见, 这种并行矩量法求解大型复杂问题时, 是能够获得比较理想的并行性能的。

表 1 不同 CPU 个数时并行矩量法性能比较

名称	CPU个数	填充时间	求解时间	总时间	并行加速比
并行 MoM	4	8min24s	153min27s	9711s	2.203
并行 MoM	9	5min26s	68min3s	4409s	

5 结束语

本文研究了在 PC 机组成的群集系统中的并行矩量法, 详细地给出了并行 CGN 方法求解矩阵方程的关键步骤, 分析了某飞机器模型散射特性, 测试了其并行性能。数值结果

表明即便是运行于非正式工作站的 PC 网络群集系统中, 棋盘块划分的矩阵填充方式以及并行 CGN 算法对于加快求解速度以及扩大 MoM 仿真规模也是非常有效的。

参考文献

- [1] Rao S M, Wilton D R, Glisson A W. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. on Antennas Propagat.*, 1982, AP-30(5): 409 - 418.
- [2] Shen Xianneng. Application of massively parallel architecture to computational electromagnetics. Tech. Report of Syracuse University, 1994.
- [3] 张玉, 苏涛, 翟会清, 梁昌洪. PC 群集系统中并行矩量法研究. *电子学报*, 2003, 31(9): 1368 - 1371.
- [4] 都志辉. 高性能计算并行编程技术[M]. 北京: 清华大学出版社, 2001: 192 - 204.

张玉: 男, 1978 年生, 讲师, 主要从事电磁场数值计算、并行电磁计算、电磁兼容分析等方面工作。

王萌: 男, 1980 年生, 博士生, 主要从事电磁场数值计算、电磁兼容分析等方面的工作。

梁昌洪: 男, 1943 年生, 教授, 博士生导师, 中国电学会会士, IEEE Senior member, 研究方向包括计算场论、计算微波、微波网络理论、电磁散射与逆散射、电磁兼容等方面。