

# EBE-PCG 算法在有限元并行计算中的应用研究

陈荣征, 李代平, 黄健, 秦昭晖  
(广东工业大学 计算机学院, 广东 广州 510006)

**摘 要:** 目前, 在研究有限元并行计算时, 讨论并行算法理论和并行算法设计与分析的居多, 研究并行算法的实现并解决实际问题较少。在 Beowulf 集群环境下, 采用 EBE 策略设计出基于 PVM 平台的 EBE-PCG 算法, 并通过一个电法勘探的典型工程算例对该算法在有限元计算中的性能进行了测试。实验结果表明, 该算法加速比和并行效率均较为理想; 在处理同等规模的问题时, 同 CG 算法、PCG 算法相比, 具有并行度更高, 耗时更少等优点。

**关键词:** 有限元法; Beowulf 集群; EBE-PCG; 并行计算; 并行虚拟机

**中图分类号:** TP301.6

**文献标识码:** A

**文章编号:** 1673-629X(2008)03-0232-04

## Research on Application of EBE-PCG Algorithm in Parallel Computing of FEM

CHEN Rong-zheng, LI Dai-ping, HUANG Jian, QIN Zhao-hui

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

**Abstract:** Currently, research on the parallel computing of finite element method, discussing parallel algorithm theory and parallel algorithm design and analysis are in the majority. However, on how to achieve and solve practical problems using the algorithm is less. In the Beowulf cluster environment, using EBE strategy to design the EBE-PCG algorithm based on PVM platform, and puts forward a typical use case in electrical surveying to test the performance of the algorithm in parallel computing of FEM. Experimental results show that the speedup and the parallel efficiency are both excellent. Compared with CG algorithm and PCG algorithm in dealing with the same scale problems, the EBE-PCG algorithm has the merits of high degree of parallelism and less running-time.

**Key words:** finite element method; Beowulf cluster; EBE-PCG; parallel computing; PVM

## 0 引言

随着计算机网络的高速发展和微处理芯片性能的不断提高, 计算机网络已经成为非常具有竞争力的并行计算载体。借助于商业化的硬件和软件, 计算机网络能够提供高性价比、高可用性的计算。这种高性能计算通常被称为集群(或称机群)计算。近年来, 集群计算已经成为并行计算领域研究的一个热点。

有限元法(FEM)被证明为科学和工程领域中连续物理系统通用的模拟方法, 但缺点是传统的串行计算机需要花费太多的计算时间<sup>[1]</sup>。FEM 需要对域进行离散化(一个网格或者模型)。FEM 的这种思想和集群分而治之的思想相似。因此, 有限元节点代码可以被并行化、在不同的基于 PVM 集群结点上运行。

EBE-PCG 算法与有限元法十分自然地联系在一起, 因为两者都是采用相似的最小化过程。

在集群环境下进行有限元并行算法的研究, 为许多工程问题的解决开辟了新的途径, 具有重要的现实意义。文中根据 EBE 策略设计出 EBE-PCG 算法, 在 PVM 环境下具体实现了算法程序的编写。在 Beowulf 集群环境下, 通过一个有限元并行计算中的典型工程算例对该算法的并行效果进行了验证, 并将该算法与 CG 算法、PCG 算法进行了对比。实验结果表明, 运用该算法处理有限元并行计算问题, 可以在算法级提高有限元并行度。

## 1 EBE 策略

EBE 策略由 Hughes<sup>[2]</sup>提出, 其基本思想可以描述如下:

假设  $M$  是一总体向量, 相应传统有限元分析中定义的单元向量  $M_k, M^k$ , 引进伪单元向量的概念。假设单元  $e$  与第  $n_1, n_2, \dots, n_e$  号总体自由度有关, 那么由

收稿日期: 2007-06-29

基金项目: 广西自然科学基金资助项目(桂科自 0229009)

作者简介: 陈荣征(1979-), 男, 山东临沂人, 硕士研究生, 研究方向为网络并行计算; 李代平, 教授, 研究方向为网络并行计算、软件体系结构。

$$M_{(k)}(i) = \begin{cases} M(i), & i = n_j, j = 1, 2, \dots, s \\ 0, & i \neq n_j, j = 1, 2, \dots, s \end{cases}$$

$$M^{(k)} = (M(n_1), M(n_2), \dots, M(n_s))^T$$

便确定了两个向量  $M_k, M^k$ , 称由此定义的  $M_k$  与  $M^k$  分别为总体坐标和局部坐标的伪单位向量。通过伪单元向量, 首先可以将  $b = kx$  的计算转化为:

$$b = kx = (\sum_e k_e) x = \sum_e (k_e x) = \sum_e (k_e x_e)$$

因此, 由于  $k_e$  与  $x_{(e)}$  均只涉及单元  $e$ , 所以不必计算  $k_e x_{(e)}$ , 而只需计算  $b^e = k^e x^e$ , 然后将  $b^e$  “对号入座” 叠加到向量  $b$  中。可见, EBE 方法将总体计算分解到单元, 无需形成总体刚度矩阵。各单元上的计算除了在一定时刻要进行一些必要的信息交换(通讯)外, 相互独立, 非常适合并行处理。

## 2 EBE-PCG 算法

### 2.1 PCG 算法

求解方程组  $Ax = b$  的 PCG 算法如下<sup>[3]</sup>:

(1) 初始化部分:

$$\textcircled{1} r_0 = b - Ax_0 \quad \textcircled{2} d_0 = Mr_0 \quad \textcircled{3} p_0 = d_0$$

(2) 每台计算节点同时执行的 PCG 算法描述为:

for  $i = 0, 1, 2, \dots$ , do

$$\textcircled{1} u = Ap \quad \textcircled{2} \alpha = r^T d / p^T u \quad \textcircled{3} x^* = x + \alpha p$$

$$\textcircled{4} \text{if } \|x^* - x\| < \epsilon \text{ then exit}$$

$$\textcircled{5} r^* = r - \alpha u \quad \textcircled{6} d^* = Mr^*$$

$$\textcircled{7} \beta = (r^*)^T d^* / r^T d \quad \textcircled{8} p^* = d^* + \beta p$$

end for

其中,  $\epsilon$  为容许收敛误差,  $M$  为预处理矩阵。

### 2.2 EBE-PCG 算法

基于多项式预处理的 EBE-PCG 算法<sup>[4]</sup>的实现过程如下:

(1) 初始化

$$\textcircled{1} \text{Initial: } x^{(e)} = 0, r^{(e)} = b^{(e)}$$

$$\textcircled{2} \text{Create } M^{(e)}: M^{(e)} = d^{(e)} \oplus \sum_{m \in \text{adj}(e)} d^{(m)} \text{ (其中}$$

$M^{(e)}$  为预处理矩阵)

$$\textcircled{3} \text{Solve the equation: } (M^{(e)}, z_0^{(e)}) = r_0^{(e)}$$

$$\textcircled{4} \text{Compute: } s^{(e)} = z_0^{(e)} \oplus \sum_{m \in \text{adj}(e)} z_0^{(m)} \text{ 和 } \gamma_0 =$$

$$\sum_{e=1}^E (r^{(e)})^T \cdot s^{(e)}, \gamma = \gamma_0$$

$$\textcircled{5} p^{(e)} = s^{(e)}$$

(2) 初始化以后, 在集群系统上, 每台计算节点同时执行的 EBE-PCG 算法描述如下:

for  $i = 0, 1, 2, \dots$ , do

$$\textcircled{1} u^{(e)} = A^{(e)} p^{(e)} \quad \textcircled{2} \frac{1}{\alpha} = \sum_{e=1}^E (p^{(e)}, u^{(e)}) / \gamma$$

$$\textcircled{3} x^{(e)} = x^{(e)} + \alpha \cdot p^{(e)} \quad \textcircled{4} r^{(e)} = r^{(e)} - \alpha \cdot u^{(e)}$$

$$\textcircled{5} \text{compute}(M^{(e)}, z_k^{(e)}) = r_k^{(e)}, \text{get } z_k^{(e)}$$

$$\textcircled{6} s^{(e)} = z^{(e)} \oplus \sum_{m \in \text{adj}(e)} z^{(m)}, p^{(e)} = (r^{(e)})^T s^{(e)}$$

$$\textcircled{7} \gamma^* = \sum_{e=1}^E p^{(e)}$$

$$\textcircled{8} \text{if } (\gamma^* < \epsilon \gamma_0) \text{ then exit}$$

$$\textcircled{9} p^{(e)} = (s^{(e)}) + (\gamma^* / \gamma) p^{(e)} \quad \textcircled{10} \gamma = \gamma^*$$

end for

上面给出的 EBE-PCG 算法反映了分管第  $e$  号单元的处理机的计算过程, EBE-PCG 算法很适合在集群环境下实现。假设计算节点的个数为  $t$ , 将所有单元分为  $t$  组, 分别记为  $P_1, P_2, \dots, P_t$  组。具体计算时, 由一个计算节点负责一组内的所有单元的相关计算。由于所有计算在单元一级上进行, 故该算法无需产生总体刚度矩阵, 只需保存单元刚度矩阵。

在网络集群环境下, 各计算节点之间的通讯时间是影响并行效率的重要因素。EBE-PCG 算法只有  $\textcircled{2}\textcircled{6}\textcircled{7}$  步需要进行少量数据传递, 故并行度非常高。

此外, 在为 PCG 法选取预处理矩阵时, 考虑到多项式预处理方法不但可高度向量化, 适合并行求解, 而且不要求刚度矩阵显示给出, 与 EBE 策略的无需形成总体刚度矩阵的特点一致, 非常适合与 EBE 策略连用。故文中算法采用多项式预处理矩阵。

## 3 算法在 PVM 环境下的实现

编程模式采用 PVM 的主/从 (Master/Slave) 模式<sup>[5]</sup>。并行处理由一个控制块 Master 和若干从属块 Slave 组成, 主/从方式中有 Master 和 Slave 两种进程, Master 主要负责接受任务、划分任务、启动计算和回收结果。每台 Slave 负责接受任务、处理任务、完成进程间的数据通信, 最后把处理结果返回给 Master。下面给出算法的实现。

### 3.1 主程序

(1) 算法。

Master 主程序

$\textcircled{1}$  接受用户输入数据。

$\textcircled{2}$  在  $X$  轴方向进行划分, 并记录各点的  $X$  坐标。

$\textcircled{3}$  向各 Slave 机子传输数据, 分派任务, 并启动 Slave 应用程序。任务分派应用二次均分, 其中  $\text{loop}[i]$  数组保存了第  $i$  台 Slave 机器需要处理的结点数。

$\textcircled{4}$  循环等待, 对每一台 Slave 机器回收计算结果。

$\textcircled{5}$  合并结果, 并计算输出。

(2) 关键代码的实现。

/\* Master 利用二次均分法对 Slave 分派任务的代码说明:

```

*/
loop = Gbdata.m_xPointNum/ProcNum; //将 X 方向的点按
处理机个数均分
for(i=0;i<ProcNum;i++) loops[i] = loop; //为每台 Slave
机分配相等的待处理点数
for(i=0;i<ProcNum - Gbdata.m_xPointNum/ProcNum *
ProcNum;i++)
loops[i]++; //将均分后剩余的点再一次分配给处理机,使
各台处理机任务大致相同
pvmnitsend(PvmDataDefault); //初始化发送缓冲区
pvm.pkint(&ProcNum,1,1); //打包变量(处理机的个数)
pvm.pkint(tids,ProcNum,1); //打包变量(为每台 Slave 机
分配 id 的数组)
pvm.pkint(loops,ProcNum,1); //打包变量(每台 Slave 机需
处理任务数的数组)
pvm.mcast(tids,ProcNum,msg_spawn); //把打包的数据以
组播形式发送给每台 Slave 机

```

### 3.2 Slave 从程序

#### (1) 算法。

①接收 Master 发送过来的数据。

②按照与 Master 一样的方法,进行 X 轴方向的划分,并记录各点的 X 坐标。

③进行 Y 和 Z 方向上的划分,记录各点的坐标和点的个数。令  $m\_pointnum = X \text{ 轴点数} * Y \text{ 轴点数} * Z \text{ 轴点数}$ ,  $m\_halfband = X \text{ 轴点数} * Y \text{ 轴点数} + X \text{ 轴点数} + 1$ , 构造大型稀疏矩阵  $PTK[m\_halfband][m\_pointnum]$ , 并初始化全置 0。

④对于测量区域中划分所得的六面体,再将每一个六面体划分为五个小的四面体,并对五个四面体的各点编号。然后对于每个四面体,求出每个四面体的单元刚度矩阵。

⑤由各个四面体的单元刚度矩阵合成矩阵  $PTK$ 。

⑥用 EBE-PCG 算法求解方程  $Ax = b$ ,  $x$  为未知量,  $b$  则根据 Slave 收到的  $loop[i]$ , 计算本地需要开始计算的初始结点,形成右端矢量  $b$ 。然后对于本机问题域上的每一点,循环  $loop[i]$  次,回代求解  $x$  值,故每一子机形成结果矩阵  $pu[loop[i]][m\_xpointnum]$ 。

⑦将  $pu$  回送给 Master。

#### (2) 关键代码的实现。

/\* Slave 客户机判断自身任务、并行回代计算各点值的代码说明 \*/

```

for (i=0;i<nproc;i++)
if (mytid=tids[i])
{
me=i;
break;
} //根据自身的 id 号,得到自身的序号 i

```

```

myloop = loops[me]; //取得自己的任务数
unsigned int NodePosition, Pos = 0;
for (j=0;j<me;j++)
Pos += loops[j]; //计算出本客户机的任务开始点在何处
for (i=0;i<myloop;i++)
{
NodePosition = Pos + i;
if(NodePosition >= Gbdata.m_xPointNum) break;
xGet(NodePosition,i); //回代计算各点值,得到矩阵 pu
[loop[i]][m_xpointnum]
}

```

## 4 算例验证及结果分析

笔者组建了一个 Beowulf 集群系统,软硬件配置如下:a. 软件环境:WPVM 3.4 on Windows XP, VC++ 6.0, TCP/IP; b. 硬件环境: P4 2.66 CPU, 512M DRAM, 100Mbps Ethernet。为了测试该算法的性能,现在以一个有限元并行计算中的具体的电法勘探算例对该算法进行两组实验。

(1)依次取不同的三维电场空间,利用 EBE-PCG 算法分别进行了串/并计算对比,对比结果如表 1 所示。

表 1 计算结果比较

序号	PC 数目	场区范围 (m <sup>3</sup> )	异常范围 (m <sup>3</sup> )	加速比	并行效率
1	5	100 * 100 * 200	10 * 10 * 10	1.56	33.2%
2	5	120 * 120 * 20	10 * 10 * 10	2.23	46.5%
3	5	150 * 150 * 20	10 * 10 * 10	3.48	75.4%
4	5	200 * 200 * 20	10 * 10 * 10	4.58	96.3%

表 1 实验结果显示,并行效率随三维电场规模的增大显著提高。原因在于:当电场的计算规模较小时,由于串行分量的计算在总的计算中占用的比重较大,加上受并行程序运行时额外开销的影响,加速比和并行效率较低;随着求解规模的逐渐扩大,串行部分所占比例缩小,并行部分逐渐增大,加速比和并行效率迅速提高。这同加速比定律的 Gustafson 修正<sup>[6]</sup>相吻合。

(2)在 4 台 PC 组成的集群环境下,分别采用 EBE-PCG 法、CG 法与 PCG 法处理同等规模的三维电场问题,各自所用时间如表 2 所示。

表 2 实验结果表明,EBE-PCG 算法同 CG 法和 PCG 法相比,在处理同等规模的三维电场问题时,所用时间更少,并行效率更高。这主要是由于该算法采用了 EBE 策略,只保存单元刚度矩阵而不集成总体刚度矩阵,从而使各计算节点之间并行度大大提高,相互之间通讯较少。此外,在运用该算法处理有限元并行计算问题时,对节点编号和单元编号以及结构的几何形状没有任何特殊要求。

表2 三种算法所用计算时间对比

PC/台	场区范围 /m <sup>3</sup>	场区电导率/ (S·m <sup>-1</sup> )	异常体大小/m <sup>3</sup>	异常体电导率/ (S·m <sup>-1</sup> )	测距/m	埋深/m	三种算法的计算时间		
							CG	PCG	EBE-PCG
4	100×100×20	1	10×10×10	20	4	4	3s	3s	3s
4	120×120×20	1	20×20×10	20	4	4	7s	6s	6s
4	150×150×20	1	30×30×20	20	4	4	21s	18s	16s
4	100×100×20	1	10×10×10	20	2	4	2min 56s	2min 34s	2min 23s
4	120×120×20	1	20×20×10	20	2	4	3min 37s	3min 3s	3min 6s
4	150×150×20	1	30×30×20	20	2	4	9min 52s	9min 13s	8min 22s

5 结 语

在 Beowulf 集群环境下,采用 EBE-PCG 算法进行有限元并行计算时,加速比和并行效率均较为理想。另外,从编程调试及运行情况来看所使用的 Windows + W PVM 网络并行平台已经相当稳定和成熟。建立的 PC 集群可充分发挥现有计算条件,成本低,效率高,非常适合进行有限元并行计算。

参考文献:

- [1] 周树荃,梁维泰,邓邵忠.有限元结构分析并行计算[M].北京:科学出版社,1997.
- [2] Hughes T J R. An Element-by-element Solution Algorithms for Problems of Structural and Solid Mechanics[J]. Comput. Meth. Appl. Mech. Engin., 1983, 36: 241-254.
- [3] 乐志华,程建钢,姚振汉.网络机群下多项式预处理 EBE-PCG 并行算法设计与实现[J].工程力学,2002,19(5):150-154.
- [4] 刘耀儒.三维有限元并行 EBE 方法[J].工程力学,2006,23(3):27-31.
- [5] 祝永志,魏榕晖,赵本立.一个基于 Windows 和 PVM 的 Beowulf 机群系统的设计与性能分析[J].计算机科学,2006,33(6):278-280.
- [6] 陆鑫达.并行程序设计[M].第2版.北京:机械工业出版社,2005.

(上接第 231 页)

指标分类	评教指标	满分	学生评教	教师互评	督导评教
教学态度	遵守教学纪律,不迟到,不提前下课	5分	3分	4分	3分
	作风严谨,教学认真,备课充分,授课流畅	5分	2分	4分	3分
	言传身教,为人师表,育人于教学之中	5分	3分	4分	3分
	经常征求学生对教学的看法或意见,耐心解答学生的疑难问题	5分	3分	4分	3分
教学内容	教学目标明确,重点突出,讲清难点、疑点	10分	5分	5分	5分
	教学内容选择适当,教学容量适中	5分	3分	4分	3分
	教学内容安排合理,理论与实践相结合	10分	5分	7分	5分
	理论内容难以适度,技能讲解、示范准确熟练	5分	4分	5分	4分
教学方法	富于启发、引导,调动学生的积极性,注重学生学习方法、思维方法的培养	5分	4分	5分	4分
	语言清晰,普通话标准,板书适当,讲课有感染力	5分	4分	5分	4分
	教学活动设计合理,练习指导得当	5分	4分	5分	4分
	了解学生的个别差异,教学方法灵活	5分	4分	5分	4分
教学效果	课堂气氛活跃,有吸引力,能激发对本学科知识的兴趣	10分	10分	10分	10分
	对教师讲授的重点内容印象深刻,能掌握或理解大部分课堂教学内容	10分	10分	10分	9分
	对学生养成积极地人生态度有很大帮助	10分	10分	10分	9分
总评:		100分	77分	82分	77分

相应指标总权重分值

图4 教师查看评教结果功能模块实现页面

- 香港:香港亚太国际出版社,1998.
- [2] 劳士健.大学生评价教师课堂教学质量的探索[J].湖州师范学院学报,2002(4):106-110.
- [3] 张立杰,张海燕,李建军.一种灰色系统的模糊 TOPSIS 方法在高校教师教学评估中干部考核工作中的应用[J].西安科技大学学报,2007,27(1):115-119.
- [4] 贾志娟,胡明生.基于网络的 Multi-Agent web 文本挖掘系统[J].微计算机信息,2006(21):266-268.
- [5] 罗万成,林红.高校教学质量评估的属性综合评价模型[J].工程数学学报,2007,24(1):187-190.