

面向对象并行有限元数值平台——ParaFe

黄玉林 袁 勇 阮红河

(同济大学地下建筑与工程系 上海 200092)

摘 要 ParaFe 是已经研发成功的并行有限元数值平台,该平台采用面向对象设计语言 C++ 及 MPI 并行库来完成整个平台的编码及调试。在 MS Window 及 Linux 两种机群环境下来测试该平台,平台运行稳定,分析结果正确,具有理想的分析效率。本文主要介绍 ParaFe 的实现思想和实现技术。

关键词 面向对象,并行计算,MPI,有限元

Numerical Platform of Object-Oriented Parallel Finite Element Method——ParaFe

HUANG Yu-Lin YUAN Yong RUAN Hong-He

(Department of Geotechnical Engineering, Tongji University, Shanghai 200092)

Abstract ParaFe, a numerical platform of parallel finite element method, has been designed successfully. For realizing ParaFe, the object-oriented language C++ and MPI (message passing interface) are employed. Using the operating system MS Windows and Linux, ParaFe is tested. The example shows that the platform can work steadily, the computing result is accurate, and effectiveness of parallel computing is reasonable. This paper mainly presents the ways and technologies for realizing the ParaFe platform.

Keywords Objected-oriented, Parallel computation, MPI, Finite element method

1 引言

用普通计算机进行大型复杂结构(如核电站,海洋平台、大型地下工程等)工程整体分析,不可避免会因为存储空间的限制导致计算无法进行和求解时间过长两大难题,实现动力响应和非线性求解问题等尤其如此。如何在合理的时间内,确保精确完成大型结构的分析,是结构分析研究人员要解决的问题,并行计算机的出现无疑是实现这一目标的有效技术保障。自 20 世纪 70 年代并行计算机问世以来,并行有限元这一全新的领域被深受关注。

并行有限元分析理论与并行计算机的发展是密切相关的,早期的并行有限元大部分都使用向量流水处理机、阵列处理机等专业并行机。自 20 世纪 90 年代,网络技术的快速增长对并行计算机的发展产生重大的影响,用网络紧密耦合多个计算节点而构成的大规模并行处理机系统—MPP (massively parallel processor),以及由多台高性能计算节点经计算机网络连接组成的机群处理环境机群并行机—COW (cluster of workstation),这些典型的分布存储计算机系统逐渐成为高性能计算机领域的两个重要研究方向。此外,1994 年 5 月发布的消息传递函数库 MPI (Message Passing Interface) 是目前所有并行计算机制造商都提供支持的一种标准或规范的代表,一个正确的 MPI 程序,可不需加任何修改即能够在所有的并行机上运行,这就给用户提供了一种可以先在低端的工作站机群上开发软件,然后在更高性能的并行计算机上运行的工作方式^[1]。

在目前的条件下搭建一个小型(16~32 个处理机)的高性能机群系统并行计算环境困难并不大,而根本问题是缺乏运行在这一环境中的有限元分析软件。目前,虽然也有支持

分布计算环境的商用有限元软件(如 MSC/NASTRAN、ABAQUS、LS-DYNA3D 等),但是价格非常昂贵。此外,这些有限元软件也不是万能的,在许多情况下,研究者们经常要自行开发有限元程序,主要着重在开发新单元、新材料模型和新分析方法等等,这样的编程工作通常要花费研究者较大的精力和时间,更何况是要开发并行有限元程序。

基于这样的背景,本课题组开发了面向对象并行有限元数值平台,该系统在分布式并行环境下运行,为研究者(用户)提供关于有限元并行策略、系统方程组的存储、结点编号优化、系统方程的并行求解等功能,使研究者只需关注物理问题本身,即研究对象的基本属性,无需在数值分析过程花过多的精力,仍然能够得到一个高效率的并行有限元分析程序。显然,这样的平台可减轻研究者的开发工作量,缩短开发时间,进而提高研究效率,有利于提高研究深度。此外,开放性和可移植性也是该平台的一个优点,前者允许用户添加新的单元类型、本构模型或分析方法等,不断扩大平台分析能力,而后者可保证所编制的程序能够在不同的分布式并行系统上运行。

2 并行策略

在分布存储并行环境下,有限元的并行策略主要有:(1)基于线性方程组并行数值求解器(Numerical Solver),对传统有限元最耗时的过程(系统方程组的求解)进行并行化以加快整个分析过程的求解时间;(2)子结构并行凝聚法,根据传统区域分裂算法(Domain Decomposition)的理论划分子域,在边界结点上建立整体结构的平衡方程,降低问题的自由度;(3)并行 EBE (Element by Element) 法,在单元级上实施有限元整体运算的并行化迭代算法,避免组集总刚矩阵;(4)其它方法,

如 FETI(Finite Element Tearing and Interconnecting)法。

目前 ParaFe 平台中实现了基于线性方程组并行数值求解器和并行子结构法两种策略。

2.1 基于线性方程组并行数值求解器

并行有限元最直接和最初的实现是利用数值并行求解器求解系统方程组,其原理如图 1 所示,即单元分析和系统方程的形成是串行过程(在一台处理机完成),而系统方程组的求解是并行过程。

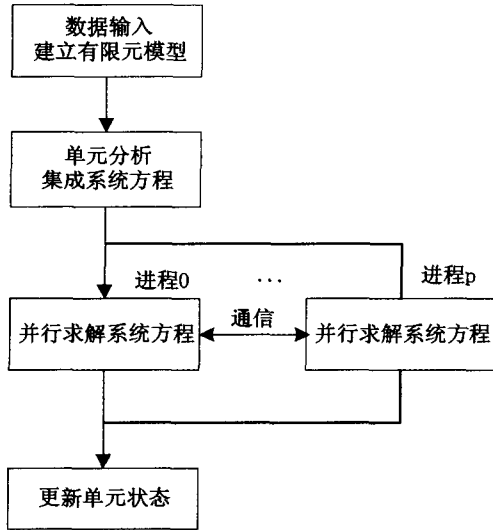


图 1 基于并行线性方程组求解流程

另一种做法是,将刚度矩阵按行进行“分割”,各处理机存储与每块范围相应的有关数据(单元、结点、荷载等),这样,单元分析和系统方程组装过程可并行进行,不需任何数据通信(当然是以少量的重复计算为代价),系统方程组的求解一般采用迭代并行求解器,如并行共轭梯度法—PCG^[2~5]。如图 2 所示,将原结构划分成若干子区域,划分面为“共享单元”(图中的阴影部分)。若每个处理机各自都保存自己子区域的相关信息(包括“共享单元”的信息),则子区域内的单元分析和总刚的集成是完全并行的,而不同处理机间有关于“共享单元”的计算是重叠的具体的计算过程如图 3 所示,显然,这种方案利用各“子结构”在单元分析和组装刚度方程过程中的独立性来实施并行化。但是,整体有限元方程的阶数仍然是原结构的全部自由度,其求解时间仍占整个分析过程较大的比例。

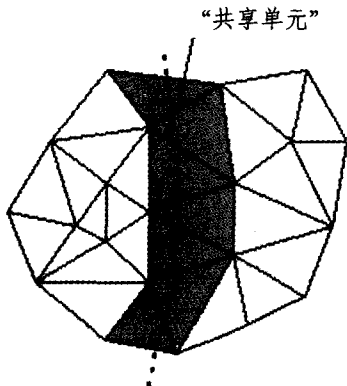


图 2 “通过单元”的区域划分方法示意图

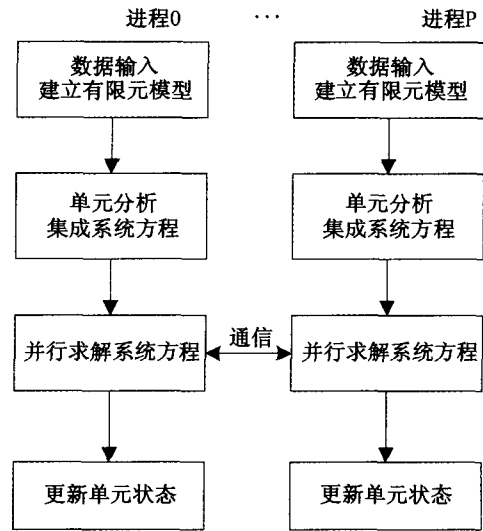


图 3 分割解算法示意图

2.2 并行子结构法

子结构法在连续域问题中常被称为区域分裂算法。并行子结构方法是根据传统串行子结构法的思想而实现的,即通过静力凝聚消去子结构的内部自由度以建立整体结构在边界结点上的平衡方程。在求得边界结点的位移后,可同时求解各子结构的内部自由度位移,并更新单元状态。由于界面自由度个数远小于整体自由度个数,因此界面方程的求解时间将大大减少,占整个分析过程的求解时间比例同样也会减少,这对于提高算法的并行性能是非常有利的。并行子结构法的求解流程如图 4,与图 3 所示的流程相比,每个进程的计算量明显加大,换句话说,计算/通信比的增加有利于提高算法的并行效率。

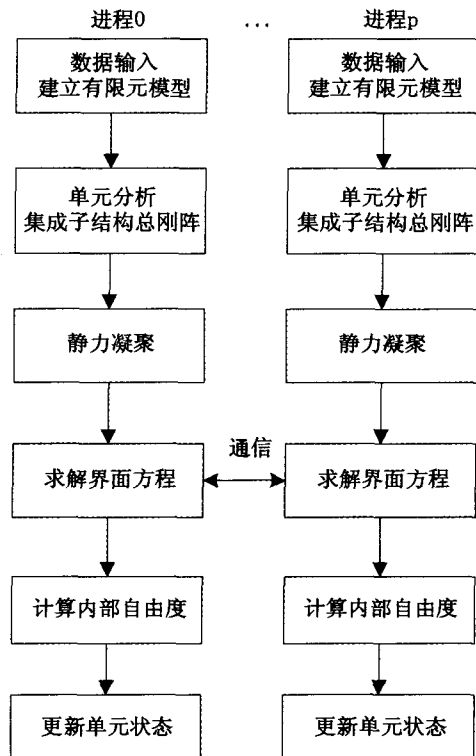


图 4 并行子结构流程图

3 MPI 支持下的并行程序编制模式

MPI 支持 SPMD 和 MPMD 两种编程模式^[6], 但 SPMD 是最常用的方式。消息传递系统通常支持两种形式的并行性开发, 一种是数据并行, 另一种是控制并行(或功能并行)。基于数据并行的思想, 可以把问题的分解归结为问题区域的分解, 即将一个大的问题区域分解为多个较小的问题区域, 然后对这些较小的问题区域并行求解来达到整个问题的求解。基于控制并行思想, 可以将一个大的问题分解成多个不同的子问题, 通过对这些不同的子问题的求解来达到整个问题的求解。实质上, 这种并行程序的设计思想是通过描述操作在处理机间的分布, 而不是数据在各处理机间的分布来开发并行。

基于数据并行的并行程序组织形式通常采用 SPMD 的方式, 即给每个处理机加载相同的程序, 但处理各自的数据。当然, 程序中可以有条件代码, 它们根据与处理机(或进程)相关的变量, 转去执行相应的语句。对于在多个处理机上运行的科学计算和工程应用软件, 为了减少管理并行指令流的复杂性, 主要采用 SPMD 的并行方式。对于这种方式的并行程序设计, 常见的程序结构有两种, 一种是主从结构, 即 Master/Slave 结构; 另一种是无主结构, 即 Hostless 结构。

基于控制并行的并行程序组织形式通常采用 MPMD 的方式, 即每个处理机执行不同的代码副本, 对数据各自完成不同的处理。对于这些应用程序, 如事务处理方面的应用程序, 由于各子功能本身所固有的异构性, 以及在由异构型机器构成的网络并行计算环境中各计算机所具有资源的异构性, 而常采用此种组织形式。

ParaFe 并行有限元平台是以面向对象的编程思想为基础, 并且是基于数据并行来组织程序, 因此采用 SPMD 方式进行编制的, 这同时也是大多数面向对象的并行程序所采用的主要方式。

4 有限元类层次设计

4.1 有限元类层次设计

有限元模型是用于描述分析问题离散化后的物理特性。该模型由结点类(Nodes)、单元类(Elements)、荷载/荷载工况类(Loads /Load Cases)及约束类(Constraints)等基本组件组成。目前, 该模型是按照有限元理论与程序设计方法的统一标准进行搭建, 具有良好的适用性和扩展性。

4.2 有限元分析类

4.2.1 系统方程组集类

系统方程组集类—Integrator 的主要功能是建立有限元的系统方程。对静力或动力响应分析来说, 该系统方程是个线性方程组, 而对结构模态分析来说, 它是个特征值方程。此外, 系统方程的组集可随分析算法的不同而不同, 如上述例子中的中心差分法、线性加速度法、Newmark- β 法和 Wilson- θ 法等组集方式都各不相同。平台中的 Integrator 类只是各抽象的基类, 而具体的组集类应根据具体的分析方法和所采用的算法来决定派生关系。

4.2.2 有限元算法类

有限元算法类—SolutionAlgorithm 对象模型如图 5 所示。具体解释参见文[7]。

4.2.3 分析类

Analysis 是个抽象的基类, 它包含 Integrator 类 SolutionAlgorithm 的对象。具体的分析方法类, 如静力分析(StaticAnalysis)类、模态分析(EigenvalueAnalysis)类、直接积分动力分析(DirectIntegrationAnalysis)类及振型迭加法(Modal-

TransientAnalysis)类等等都是 Analysis 的子类。这些分析类的主要功能是在最高层次上控制分析流程。

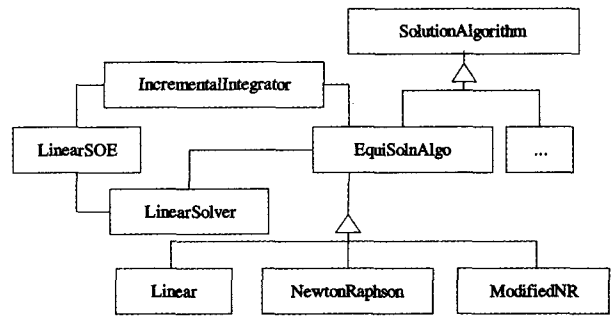


图 5 ParaFe 的 SolutionAlgorithm 对象模型

5 有限元并行计算

5.1 并行有限元模型建立

该平台目前没有前处理建模工具, 程序通过读取用户提供的数据库文件来创建有限元模型。在分布存储并行环境下, 可通过以下两种方法进行建模。

5.1.1 主机/节点模式

图 6 为主机/节点模式的示意图, 运行在主机上的主进程先读取数据文件, 再对模型进行初始化, 最后将相应数据分配到相应的进程上。与无主机模式相比, 它的优点在于可避免各进程在初始化工作中的重复, 但其缺点则更为突出, 在从主机发送相应数据到其它进程的过程中, 要传递大量的对象数据(单元、结点、荷载、约束等数据), 并需要接收进程根据传来的信息创建相应的对象。这类对象的发送属于不连续数据发送, 因此, 编程工作量很大, 而且每次添加新对象或修改对象的属性都需要修改相应的通信代码。

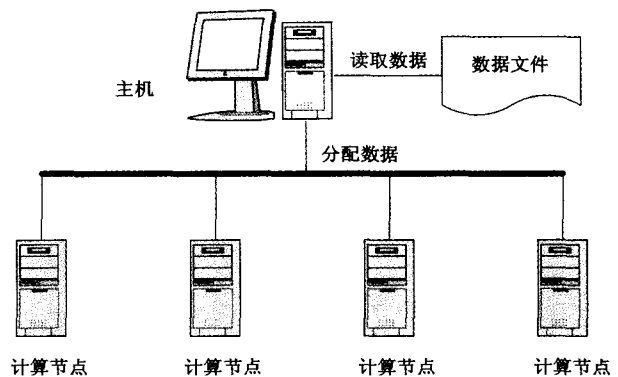


图 6 主机/节点模式的输入方法

5.1.2 无主机模式

图 7 为无主机模式的示意图。在该模式中, 各进程同时读取同一份数据文件(或一份数据文件的不同拷贝), 并建立进程与子结构之间的对应关系, 再进行初始化, 最后删除与本进程无关的数据。

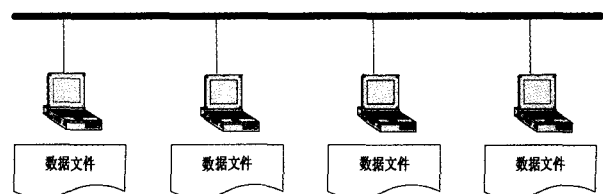


图 7 无主机模式的输入方法

显然,采用无主机模式的好处在于各进程间不需通过通信也能够形成自己所需的模型信息,并且能避免复杂的不连续数据通信,而且读取数据的代码与串行子结构程序完全相同。因此,本平台采取了这种模式来建立并行有限元模型。

5.2 并行有限元求解器

并行求解器是并行有限元平台开发的重点和难点,相对工作量比较大。

无论采用何种并行求解器,各进程在进行并行求解过程中都需要通过互相通信来交换数据。与串行程序的编制和调试相比,并程序的编制与调试要复杂得多(一般不能通过设置断点进行调试,而只能通过打印信息进行调试),因此,设计一个简洁的并行代码显得非常重要,这不仅有利于程序的调试,还可以减少错误的产生。本平台采用的数据通信通道技术能有效体现这一特性。

6 算例

对如图 8 所示的悬臂梁结构进行损伤分析,梁的截面尺寸和配筋如图 8。将该梁划分为 2000 个单元,6321 个节点,荷载划分为 20 步逐步施加。

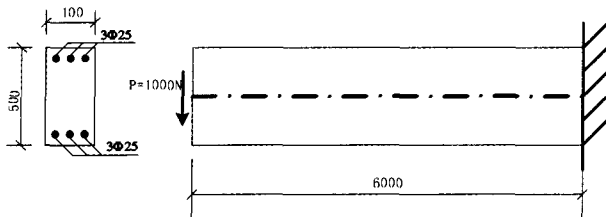


图 8 算例示意图

进行并行分析时,分别将该梁划分为 2 个子结构和 4 个子结构,计算环境为由 4 台 PC 组成的机群(每台配置 AMD1.8GHzCPU、256MB 内存、window2000 操作系统,网络传递速度为每秒 100MB)。计算分析的时间见表 1。

表 1 串行程序及并行程序的分析时间对比

比较对象	一步迭代的时间分配(秒)				总分析时间 (190 迭代步)
	formK	formF	solve	update	
串行	0.26	0.20	3.24	83.70	4h40m
2 台处理机	0.14	0.10	$6e^{-4}$	39.76	2h15m
4 台处理机	1.66	0.05	0.013	17.90	1h8m

从表 1 中可以看出,每一迭代步中的 update 部分占总分析时间最大的比例(基本大于 90%),主要花在计算各单元高斯点上损伤系数的过程中。经并行处理及实现计算与通信重叠后,这一过程所需的时间明显减少,从而降低了整个分析过程所要的时间(2 台处理机和 4 台处理机的计算时间均达到

120 理想的加速比),明显地展示出并行计算的效果。

结论 面向对象并行有限元数值平台具有良好扩充性,并且维护非常方便。该平台除了为用户提供一个健壮的有限元体系外,还为用户提供多种优化的系统方程存储方法、多种高效的线性方程组求解器,以及丰富的数学工具,可帮助用户快速地开发串行或并行有限元程序,最大限度减轻用户的开发工作量,使用户把更多的精力集中在开发与其研究目的有关的单元和材料模型上。

本平台采用了两种并行策略,以 MPI 并行程序设计函数库为并行支撑环境,实现了面向对象有限元并行计算功能。对想用并行程序的用户来说,他们可不需了解并行编程,所作的开发工作与串行程序完全一样,而并行计算功能由平台提供。从而,极大地方便了用户的使用。

对于该平台,本课题组目前正在进行和今后将要进行的工作有:

(1) 丰富平台的单元类型和材料模型

该平台已开发了梁杆单元、平面单元、板单元等,但单元的材料本构模型需要进一步增强;

(2) 增加新的分析方法

传统的非线性分析方法如增量法、迭代法(Newton-Raphson、修正 Newton-Raphson 法)和混合法一般只能分析材料的上升段,而不能分析材料的下降段。在这种情况下,需要开发其它更有效的分析方法如逐步搜索法、虚加刚性弹簧法、位移控制法、弧长法等;

(3) 增加新的并行策略及开发更高效的并行求解器

当在较小的并行环境下(处理机节点数较少)进行大规模分析时,本分析平台有可能因单机内存的限制导致计算效率不理想。在这种情况下,并行多重子结构法是非常好的解决方案。

参考文献

- 阮红河,袁勇,柳献.分布存储环境并行有限元研究进展[J]. 同济大学学报(自然科学版), 2005, 33(1):21~27
- Khan A I, Topping B H V. Parallel finite element analysis using Jacobi-conditioned conjugate gradient algorithm[J]. Advances in Engineering Software, 1996, 25:309~319
- 曹骥. MPI 环境先进有限元程序方法研究[D]. 上海同济大学建筑工程系, 2002
- 袁勇,曹骥. 机群先进计算机的结构分析平台[J]. 结构工程师, 2003, 66:23~329
- Papadrakakis M, Bitzarakis. Domain decomposition PCG methods for serial and parallel processing[J]. Advances in Engineering Software, 1996, 25:291~307
- 都志辉. 高性能计算机并行编程技术—MPI 并行程序设计[M]. 清华大学出版社, 2001
- 阮红河. 结构分析理论的并行有限元方法[D]. 上海同济大学建筑工程系, 2004
- 和力,鱼滨,和燕,吴丽贤. 一种用 UML 开发组件式 Web 应用系统的过程[J]. 计算机工程, 2005, 31(1):95~97
- 王少锋. 面向对象技术 UML 教程[M]. 北京:清华大学出版社, 2004
- 埃里克森(美),等著. 余安萍,等译. UML2 工具箱[M]. 北京:电子工业出版社, 2004
- MediaWiki 引擎源码下载[EB/OL]. <http://www.cncode.com/downinfo/121.html>, 2005, 8
- 天下维客[EB/OL]. <http://www.allwiki.com/>, 2005, 9
- 周莹新,艾波. 软件体系结构建模研究[J]. 软件学报, 1998, 9(11):867~872

(上接第 253 页)

弱。因此,可以研究将形式化语义描述能力强的层次状态图、Petri 网、进程代数、体系结构描述语言 ADL 和 UML 结合起来描述软件体系结构,以弥补 UML 非形式化的不足^[2]。

参考文献

- 李玉萍,毛少杰,张桂林. 面向对象技术和 UML 在剧情产生系统中的应用[J]. 系统仿真学报, 2005, 17(6):1311~1314
- 戎玫,张广泉,刘艳. 基于软件体系结构和 UML 的图书管理系统设计与实现[J]. 计算机科学, 2005, 32(6):224~227