文章编号：0253-374X(2016)07-1121-09 DOI:10.11908/j.issn.0253-374x.2016.07.022

向量式有限元桁架结构并行程序节点分配技术

杜庆峰，吴 瀚
(同济大学 软件学院，上海 201804)

摘要：结合向量式有限元(VFIFE)的计算规则以及桁架结构的特点，提出了一种并行程序节点分配机制. 通过对桁架结构模型数据的分析，定义模型数据的分解规则，动态实现对模型数据的分解. 依据分解的结果来动态划分并行计算的数据集，并且基于特定的并行计算框架完成并行计算. 实例验证表明，该节点分配机制是有效的，并且极大地提高了计算效率.

关键词：向量式有限元(VFIFE)；桁架结构；并行算法
中图分类号：TN919 **文献标志码**：A

Node Distributing Technology for Parallel Computing of Spatial Truss Structure Using Vector Form Intrinsic Finite Element

DU Qingfeng, *WU Han*
(School of Software Engineering, Tongji University, Shanghai 201804, China)

Abstract: After studying the computation rules in vector form intrinsic finite element (VFIFE) and the feature of spatial truss structure, a node distributing mechanism for parallel computing was proposed. According to the defined rules of decomposing, the model data of spatial truss structure were dynamically decomposed. Based on the results of decomposition, the data sets of nodes were dynamically decomposed, and the parallel computing was accomplished in the specific frame. The numerical experiments indicate that, the node distributing mechanism greatly improves the speed of spatial truss structure analysis, in contrast with those based on serial computing or classical finite element method.

Key words: vector form intrinsic finite element (VFIFE); spatial truss structure; parallel computing

1 研究背景及问题的提出

1.1 背景介绍

随着计算机科学的迅速发展，计算科学、分析理论和物理实验构成了现代科学发展的三大支柱. 在建筑工程领域，结构力学、有限元理论是目前主流的分析架构，这种架构主要包括两部分：为了描述结构体的性质和物理行为设定一组描述参数；为了进行数值分析而提出的一组行为变化准则及简化假设. 依据此架构，工程师们可以通过数学方法来模拟一个结构，规划计算流程，得到结构上任意一点的位置变化，以及其他力学参数.

向量式有限元(VFIFE)[1-3]是由美国普渡大学丁承先教授提出的结构行为分析的一个创新概念. VFIFE是求解结构的大变形、大变位、弹塑性、碰撞、倒塌等非线性或不连续力学行为的新方法. 以此理论为基础，可以通过简单的而系统化的计算程序对结构的真实行为进行仿真.

VFIFE的研究与应用尚处于起步阶段，对许多实际问题的模拟还比较初步. 尽管如此，在许多方面的应用尤其是涉及多个结构体的系统动力、大变形等问题的处理上，VFIFE已经显示出了独特的优越性，即程序的稳定性、计算的收敛性与准确性、程序编写的简洁性以及适合并行处理等一系列优点，且无需集成整体刚度矩阵[4]，具有更高的程序开发效率.

空间桁架结构是建筑工程领域常见的一种格构化梁式结构. 常用于大跨度的厂房、展览馆、体育馆和桥梁等公共建筑中. 本文将以桁架结构为研究对象，实现VFIFE程式计算过程的并行化处理.

1.2 VFIFE计算过程的不足及问题的提出

1.2.1 VFIFE计算过程

VFIFE计算过程分可为3个模块：前处理模块、

收稿日期：2015-09-27
基金项目：国家自然科学基金(41171303)
第一作者：杜庆峰(1968—)，男，教授，博士生导师，工学博士，主要研究方向为软件测试、计算机相关学科交叉领域.
E-mail:du_cloud@tongji.edu.cn

程式计算模块、后处理模块. 结构体的模型数据通过前处理模块生成并解析,输入到程式计算模块进行计算,最终输出可由后处理模块进行展示的行为数据. 流程图如图1所示.

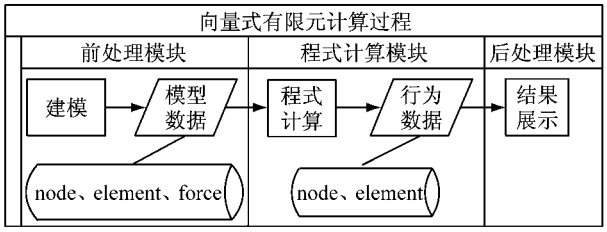

图1 VFIFE计算流程图

Fig.1 Calculation flowchart of VFIFE

前处理模块中,通过建模生成的数据即VFIFE程序计算的输入数据,由第三方建模软件SAP2000导出的模型文件转换而来. 以桁架结构为例,桁架是一种由杆件在两端用节点连接而成的结构,因而模型数据主要包括三部分:node、element、force,分别包含了节点信息、杆件与节点的拓扑信息以及外力信息.

模型数据的节点信息node每一行代表一个节点,包含了节点的编号值、节点的类型(铰接点或刚节点)、节点的坐标信息以及节点的质量,例如:

nodeId	type	posit_x	posit_y	posit_z	mass
0	2	0.000 000 000 000 000 0e+000	0.000 000 000 000 000 0e+000	0.000 000 000 000 000 0e+000	10.0
1	2	6.000 000 000 000 000 0e+000	0.000 000 000 000 000 0e+000	0.000 000 000 000 000 0e+000	10.0

模型数据的杆件信息element每一行代表一个杆件,包含了杆件的编号值、杆件两端节点编号值以及杆件的一些力学参数(杨氏模量、横截面积、密度和极限力),例如:

elementId	nodeIdA	nodeIdB	youngModulus	area	density	extremeForce
0	0	4	100.0	1.0	1.0	10.0
1	1	5	100.0	1.0	1.0	10.0

模型数据的力学信息force每一行代表一个作用力,包含了力的编号值、类型(默认为集中力),力的作用位置、方向以及作用时间,例如:

forceId	type	posit_x	posit_y	posit_z	direc_x	direc_y	direc_z	directMag	elementId	startTime	endTime
0	1	6.000 00e+000	6.000 00e+000	1.500 00e+001	0	1	0	1 000.0	37	0.0	50.0
1	1	0.000 00e+000	6.000 00e+000	1.500 00e+001	0	1	0	1 000.0	39	0.0	50.0

程式计算模块中,计算程序依据桁架结构模型的位移迭代公式编写,将前处理模块生成的模型数据作为第一帧输入,由节点当前帧以及上一帧的位置计算出该点下一帧的位置. 该位移迭代公式可简写为

$$\boldsymbol{X}_{n+1} = C_1\boldsymbol{X}_n + C_2\boldsymbol{X}_{n-1} + C_3\boldsymbol{F}_n \tag{1}$$

式中:$\boldsymbol{X}_{n+1}$、$\boldsymbol{X}_n$及$\boldsymbol{X}_{n-1}$依次为节点下一帧坐标向量、节点当前帧坐标向量以及节点上一帧坐标向量;C_1、C_2及C_3均为与结构模型本身相关的力学常量;$\boldsymbol{F}_n$则表示节点当前帧所受的外力、作用力等.

后处理模块中,行为数据即程序计算的输出数据,由程式计算程序直接导出,描述了外力作用下结构模型的变形行为,分为两部分:时间信息的几何模型数据Node.txt和包含时间信息的拓扑模型数据Element.txt.

行为数据中几何模型数据包括了帧、节点编号值、节点的坐标信息,例如:

time	nodeId	posit_x	posit_y	posit_z
1.000 000 000 000 000e−002	2.200 000 000 000 000e+001	6.000 000 000 000 000e+000	6.000 000 000 000 000e+000	1.500 000 000 000 000e+001
1.000 000 000 000 000e−002	2.300 000 000 000 000e+001	0.000 000 000 000 000e+000	6.000 000 000 000 000e+000	1.500 000 000 000 000e+001

行为数据中拓扑模型数据包括了帧、杆件编号值、杆件两端的节点编号值,例如:

time	elementId	nodeIdA	nodeIdB
1.000 000 000 000 000e−002	5.100 000 000 000 000e+001	1.100 000 000 000 000e+001	1.200 000 000 000 000e+001
1.000 000 000 000 000e−002	5.200 000 000 000 000e+001	1.200 000 000 000 000e+001	1.700 000 000 000 000e+001

后处理行为数据可直接用于展示结构模型的动画效果,后处理模块的程序可以根据每一帧的行为数据绘制出模型,并连续地展示出来. 行为数据规模与结构模型的大小、演示时间帧数量成正比.

1.2.2 问题的提出

现阶段,前处理模型数据均来源于传统的有限元分析工具(SAP2000、ANSYS等)建模导出的模型数据文件,然而这些建模软件封装度高,没接口支持,算法灵活性差,导出的模型数据文件无法直接用于VFIFE程序计算.

在程式计算模块,程序代码处理海量数据时,仍然是逐点依次进行迭代运算,需要等所有节点一次迭代完成,再进行下一次迭代运算,因而计算效率不高,且无法体现VFIFE适合并行处理等优势.

在后处理模块，后期的数据读取与行为分析显示也已经完成，通过调用 OpenGL 库，整个有限元分析过程可以以三维动画的形式展示在用户面前[5]. 此外，杜庆峰等[5-6]提出的压缩模型及算法为数据文件的云端存储提供了可能，具有很高的实用性.

针对以上不足，有必要进行 VFIFE 行为数据的并行分布式处理研究，这对于推广 VFIFE 这一新型结构力学方法，实现高性能的并行有限元分析系统具有重要意义. 本文以建筑工程中常见的大型复杂桁架结构为例，通过分析 VFIFE 前处理模型数据结构，在云计算环境下实现程序计算过程的并行分布式处理，并与传统有限元处理方式[7-8]的结果进行对比.

2　并行计算节点分配模型

2.1　节点标识规则及节点任务分配

2.1.1　相关定义

在 VFIFE 分析中，空间几何实体经离散化处理，成为有限元模型[9]，然后按一定的规则将所有杆件和节点标识赋予唯一的编号值，通过节点的运动状态（位置、速度、加速度）来描述结构的运动状态[10]，并基于此进行分析和计算. 尽管标识的顺序原则上不影响计算结果，可以是任意的，但节点标识的顺序对并行计算节点分配起着决定性的作用.

节点编号值排序的大部分方法是将问题转化为有限元网格所生成的各种图的顶点排序问题[11]. 传统有限元进行节点标识排序是为了保证有限元整体刚度矩阵的带宽最小，向量式有限元虽然无需集成刚度矩阵，但为了方便并行计算前对整体结构的划分，应该参照传统有限元的节点标识方式.

本文针对复杂桁架结构分析提出的节点标识方式参考了姜涛等[12]提出的优化算法. 相关定义如下：

定义 1　相邻节点、节点的度与节点熵

在桁架结构中，由若干杆件围成的封闭图形称为一个单元 r，在同一单元 r 内的所有节点互称为相邻节点.

假设任一节点 u 的所有相邻节点总数为 d 个，称为节点 u 的度，则节点 u 的相邻节点集合表示为 $S_u=\{s_1,s_2,s_3,\cdots,s_d\}$.

节点 u 所在的单元均称为该节点的相关单元，假设节点 u 的所有相关单元总数为 j 个，则节点 u 的相关单元集合表示为 $R_u=\{r_1,r_2,r_3,\cdots,r_j\}$，该节点 u 的节点熵 $\delta=d/j$.

定义 2　桁架结构层次展开树

假设在基于 VFIFE 分析的某一桁架结构中，其中某一节点 P 的度最小，将其作为首节点，再将结构中其他节点按与首节点 P 的相邻关系分层，形成一个展开树. 从根节点 P 开始为第 1 层，其相邻节点为第 2 层，第 2 层的所有节点的相邻节点为第 3 层，依次类推直到所有节点都编入树内，则定义首节点 P 产生的展开树 $L_P=\{L_1,L_2,\cdots,L_i,\cdots,L_G\}$，$i$ 为层号，G 为该树的总层数. 记第 i 层的节点总数为 $|L_i|$，则定义展开树 L_P 的宽度 $w_p=\max\{|L_i|,i=1,2,3,\cdots,G\}$.

定义 3　直径端点

假设在定义 2 的桁架结构层次展开树 L_P 中，其首结点为 P，尾节点为 Q，可使展开树树宽 w_p 最小，则称 P 和 Q 为该结构的直径端点.

由于计算直径端点的迭代过程较为繁琐，依据桁架结构的特殊性，可近似地取空间坐标中距离最远的 2 个点为直径端点（不影响问题的研究）. 假设直径端点 P 和 Q 的空间坐标分别为 $P(x_p,y_p,z_p)$ 及 $Q(x_q,y_q,z_q)$，则应满足 P 和 Q 2 个点之间的距离 $l_{pq}=\sqrt{(x_p-x_q)^2+(y_p-y_q)^2+(z_p-z_q)^2}$ 为最大值.

定义 4　合成展开树

假设构造以直径端点 P 和 Q 为根的桁架结构层次展开树 $L_P=\{L_{P1},L_{P2},L_{P3},\cdots,L_{PG}\}$ 以及 $L_Q=\{L_{Q1},L_{Q2},L_{Q3},\cdots,L_{QF}\}$，然后按照姜涛等[12]提出的算法合并树 L_P 和 L_Q 形成新的合成展开树 $L=\{D_1,D_2,D_3,\cdots,D_k\}$[13]，记树 L_P 和 L_Q 的宽度分别为 w_p 和 w_q，则合成展开树 L 的树宽满足条件 $w\leqslant\min\{w_p,w_q\}$.

定义 5　模型文件节点数据集

以桁架结构为例，前处理模型文件中，将节点信息表示为数据集 $\mathrm{node}(N,T,\rho,M)$，简称为节点数据集，其构成元素含义为：$N=\{n_0,n_1,n_2,\cdots,n_n\}$ 表示节点的编号值集合，集合中的 n_i 表示每个节点的编号值（Id），是由 0 至 n 的整型数据；$T=\{a_0,a_1,a_2,\cdots,a_n\}$ 表示节点的类型集合，集合中的元素 a_i 表示每个节点的类型，$a_i=0$ 表示该节点类型为自由点，即无约束的点，$a_i=1$ 表示该节点的约束类型为铰接，$a_i=2$ 表示该节点的约束类型为刚接；$\rho=\{\rho_{nx},\rho_{ny},\rho_{nz}\}$ 表示节点坐标信息集合，其中 $\rho_n=\{\rho_{n0},\rho_{n1},\rho_{n2},\cdots,\rho_{nm}\}$，集合中的 ρ_{ni} 表示每个节点的三维空间坐标值；$M=\{m_0,m_1,m_2,\cdots,m_n\}$ 表示节点的质量集合，集合中的 m_i 表示每个节点的质量.

定义6 模型文件杆件数据集

以桁架结构为例,前处理模型文件中,杆件信息表示为数据集 element(E,N_A,N_B,C),其构成元素含义为:$E=\{e_0,e_1,e_2,\cdots,e_H\}$表示杆件的编号值集合,集合中的 e_i 表示每个杆件的编号值;$N_A=\{A_{n0},A_{n1},A_{n2},\cdots,A_{nH}\}$,集合中的元素 A_{ni} 表示每个杆件某一端的节点编号值,显然 $A_{ni}\in\mathbf{N}$;$N_B=\{B_{n0},B_{n1},B_{n2},\cdots,B_{nH}\}$中的元素 B_{ni} 表示杆件另一端的节点编号值,同理有 $B_{ni}\in\mathbf{N}$;数据集中的 C 表示该杆件的一些力学常数,包括杨氏模量、杆件的截面积及密度,以及杆件的极限应力,这些力学常量均为程式计算过程中所需参数. 将杆件数据集 element 中的任一行表示为 $l_e(e,N_a,N_b,C)$,即编号为 e 的杆件.

2.1.2 节点标识及分配规则

假设在任一桁架结构模型文件中,首先依据定义3选取模型文件节点数据集中距离最远的2个点 P 和 Q 作为直径端点,然后以这2个端点作为首节点生成该桁架结构的层次展开树 L_P 和 L_Q. 依据定义4合并为合成展开树 L. 对于合成展开树 $L=\{D_1,D_2,D_3,\cdots,D_k\}$,标识连续的正整数给 D_1 上的节点,用最小编号值0标识展开树的起点 P,设 μ 为 D_1 中已标识节点中最小的,则按照节点熵 δ 增加的顺序对 μ 的相邻节点进行标识. 迭代上一步骤完成本层大部分节点标识,对于本层的剩余节点,选择最小的节点编号值重复迭代过程,直到本层所有节点都被标识. 从根节点层到最后一层依次重复上述算法,完成整体桁架结构的节点标识.

对于杆件的标识,同样依照节点编号值顺序来进行编号,从编号值最小的节点 P 开始,在其相邻节点中查找节点编号值最小的点,将最小编号0分给连接这2个点的杆件,然后再按照节点编号值增加的顺序分配连续正整数到以节点 Q 为端点的杆件. 依次对其他节点进行上述过程,若遇到已经被标识的杆件则保留原编号,继续查找其他相邻点,直到结构中所有杆件均被标识.

在不考虑断裂的情况下,杆件与节点的拓扑数据,即模型文件中的杆件数据集 element 不会随时间变化发生改变,因而在并行计算前,只需对桁架结构的节点数据集 node 进行分配即可. 结合以上特点,针对并行计算过程中节点分配模型,提出以下定义:

定义7 基准节点与分区

假设需计算的桁架结构模型节点总数为 λ 个,将要划分成 k 个分区进行并行计算,各个分区的编号取1至 k 的正整数,则称 $\beta=\lceil\lambda/k\rceil$ 为各个分区的理想分配节点数,该数学符号含义为向上取整数. 假设首节点 P 的编号值为0,则将编号值为 $b_i=\beta_i(i=1,2,\cdots,k)$ 的点定义为基准节点. 如果 $b_k=\beta_k>\lambda$,则编号值为 b_k 的节点应该取尾节点 Q. 将所有基准节点的集合记为 $B\{b_1,b_2,b_3,\cdots,b_k\}$.

定义8 分割杆件与分割节点

在程序计算的位移迭代公式中,是以各个节点为单位,迭代计算出节点在每一帧的位置,迭代公式(1)中,除了节点合力 $\boldsymbol{F}_n$ 中的节点内力 f_n 需要根据当前帧的杆件长度计算外,其他数据均来源于定义5中的节点数据集 node. 而计算某一节点的内力 f_n 需要先计算所有与该节点相连的杆件的长度变化,即需要杆件另一端节点的位置信息. 在并行计算过程对输入数据进行划分时,应保证各个分区的整体性,即各个分区中的节点是连续的,而相邻的分区通过若干杆件相连,这种杆件定义为分割杆件,显然分割杆件的数量越少,并行计算时需要在各个分区外获取的数据量越小. 分割杆件两端的节点均称为分割节点.

在上述节点重新标识的基础上,依据定义7和8,得出节点分配应遵循的原则:为了保证并行计算时负载的均衡,需尽量保证各个分区中分配到的子节点数据集大小相近. 为了使程式计算的迭代过程顺利进行,需标记出各个分区中包含在分割杆件中的点,这些点需要随着每一次迭代而更新,并运用于2个或以上的分区.

2.2 节点分配模型

根据第2.1节中的相关定义,本节给出具体的节点分配模型.

假设有一个桁架结构模型文件 θ 需要分配到 k 个分区中参与并行计算,基于定义5中的节点数据集 node 及定义6中的杆件数据集 element,模型数据文件的构成可以表示为 $\theta(F,\text{node},\text{element})$,其中 $F(\text{Id}_F,W,T)$ 为模型数据文件中的外力元素,此集合中的元素 Id_F 表示此外力的编号值,是由0至 n 的整型数据. 元素 W 表示集中外力作用的3个要素信息,即力的大小、力的作用点及力的方向. 集合中元素 T 表示力的斜坡加载时间参数,包括斜坡加载的开始时间、结束时间等,T 的大小直接关系到程式计算生成的后处理行为数据文件大小.

基于以上分析,节点分配模型逻辑如下:

步骤1 应依据定义4的合成展开树以及标识规则对模型的节点以及杆件进行重新标识,得到重

新标识后的节点数据集 node 与杆件数据集 element.

步骤 2　计算出节点数据集 node 的 k 个基准节点，依据定义 7 对基准节点的定义，得到基准节点的集合 $B\{b_1,b_2,b_3,\cdots,b_k\}$.

步骤 3　依据各个基准点编号值将节点数据集 node 划分为 k 个子节点数据集，通过遍历数据集 node 中的各个节点编号值 n_α，将满足条件 $n_\alpha\in[b_i,b_{i+1}](i=1,2,3,\cdots,k)$ 的节点集合表示为子节点数据集 $\text{node}_i'(N',T',\rho',M')(i=1,2,3,\cdots,k)$，由此得到 k 个子节点数据集，显然有 $\text{node}_i'\in\text{node}(i=1,2,3,\cdots,k)$，这些子节点数据集将分配到 k 个分区中进行并行计算.

步骤 4　在任一子节点数据集 node_i' 中，遍历其中所有节点，对任一编号值为 n_α 的节点 α，在杆件数据集 element 中查找满足编号值 $n_\gamma\in l_e(E,n_\alpha,n_\gamma,C)$ 的节点 γ，标记为节点 α 相邻节点，其中 l_e 为杆件数据集 element 中的某一行信息. 依据定义 8 中对分割杆件与分割节点的定义，如果节点 α 和它的某一个相邻节点 γ 满足条件 $n_\alpha\in[b_{i-1},b_i]$ 且 $n_\gamma\notin[b_{i-1},b_i]$，则标记节点 α 和 γ 为分割节点，假设节点 α 所处分区编号值为 τ，节点 γ 所处分区编号值为 ω，则节点 α 与节点 γ 均应标明所处分区编号值对 $\xi(\tau,\omega)$，因而可得到分割节点数据集 $\text{node}_\xi(N,T,\rho,M,\xi)$. 分割节点应当按照步骤 6 进行特殊处理，保证分配后的节点数据集在并行计算过程的连贯性.

步骤 5　依据迭代式(1)，在程序计算过程中，外力元素数据集 F 只随迭代次数而发生变化，而杆件数据集 element 仅表示模型中杆件与节点的拓扑关系. 故数据集 F 以及 element 不会依据每一次迭代的结果而更新数据，无需参与分块过程，仅需对节点数据集 node 进行分块. 并行计算过程中，各计算节点共享数据集 F 与 element.

步骤 6　在计算过程中需要将分割节点数据集 $\text{node}_\xi(N,T,\rho,M,\xi)$ 储存在共享缓存中，依据分割节点的分区编号值对 $\xi(\tau,\omega)$ 来选择计算时需要获取数据所在的分区，并随计算过程的每一次迭代而更新当前帧的数据.

步骤 7　基于上述分配的结果执行并行计算过程，得到计算结果，依据子节点数据集 node_i' 中的各节点编号值整合结果. 因为每个节点的编号值是唯一的，因而依照节点编号值 N' 递增的次序再次排序，即可得到每一帧的节点数据结果，每一帧的外力元素数据集 F 与杆件数据集 element 未进行分块，故无需整合.

3　分配模型算法及并行处理验证

根据第 2.2 节中节点分配模型的逻辑，本节将在 Spark 这一云计算平台下实现 VFIFE 复杂桁架结构程序计算的并行化，并使用资源管理器 YARN 进行任务的调度. Spark 的编程模型与 Hadoop 的 MapReduce 编程模型非常类似，但是 Spark 中的容错并行数据结构(RDD)模型使得 Spark 可以实现数据集的相互转化和缓存等操作，这在处理迭代计算方面具有非常大的优势，因而成为本实验的首选.

3.1　分配模型算法

并行处理以桁架结构的模型数据文件 $\theta(F,\text{node},\text{element})$ 为对象，其存储方式为多维数组，每一个数组元素表示一个外力元素信息或是节点、杆件的位置信息. 首先对模型文件中的节点数据集 $\text{node}(N,T,\rho,M)$ 按照本文的节点标识规则进行重新标识，在重新标识后即依据需要分成的分区数对节点进行分块，并将每个分区中的分割节点信息保存到 RDD 中，在每一次迭代完成后，需将上一帧中的节点信息同步到各个分区中，进行下一步迭代计算.

在 Spark 的编程模型中，核心计算部分引入了 RDD 的基于内存的 MapReduce，底层依赖于 Hadoop 分布式文件系统(HDFS)和 YARN/Mesos，因而本文中程序计算的并行分布式处理大致划分为 2 个阶段执行：Map 和 Reduce. Spark 提供了一种共享变量——广播变量，可供所有机器获取其中数据，利用这个机制可以在所有 Slave 节点上共享只读数据. 结合第 2.2 节的分析，本实验将输入文件中的数据集放入到广播变量中，以供所有 Slave 节点读取.

Spark 的 RDD 可提供 2 种操作，即转换以及动作. 转换可从 RDD 生成新的 RDD，动作则能够将 RDD 数据集的执行结果写入存储系统或者传回驱动程序. 在本实验的各个分区中，将并行地执行程序计算中的位移迭代公式，对各个分区中的模型文件节点数据集进行计算，而在计算当前帧时，会将当前帧和上一帧的节点位置信息保存到 RDD 中，并且对各帧的 RDD 进行缓存，以便下一步迭代的执行，同时还要写到输出的行为数据中作为记录. 缓存的 RDD 一般存储在内存中，如果内存不够，可以写到磁盘上[14].

综上所述，基于 Spark 的向量式有限元复杂桁

架结构并行分布式计算流程大致如图2所示.

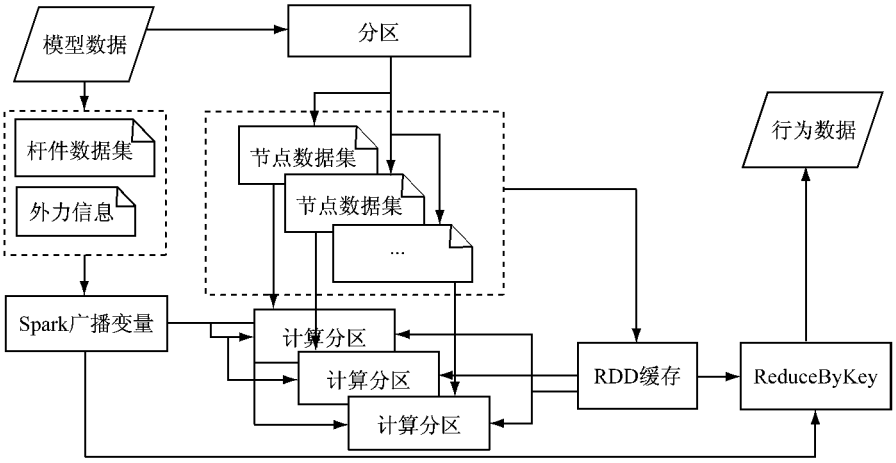

图2 并行计算的流程图

Fig.2 Flowchart of parallel computing

结合上图,向量式有限元并行处理的步骤如下:

Driver:基于Spark的底层驱动类,可通过方法函数setMapper()、setCombiner()和setReducer()驱动Mapper类、Combiner类和Reducer类,从而在相应的类中实现Map、Combine以及ReduceByKey等操作函数对数据集进行处理.

Mapper:集群启动后,各个节点会采用Spark中默认的textFile()类将分区中的子节点数据集node_i'输入文件的行以<Key, Value>的形式输入到map函数,这里的Key为各个节点的编号值$N'=\{n_0,n_1,n_2,\cdots,n_n\}$,Value为其他文本数据.分类后的各个数据块可依据编号值表示为:$N_1'=\{n_1,n_2,\cdots,n_p,n_{p+1},\cdots,n_q\}$, $N_2'=\{n_p,n_{p+1},\cdots,n_q,n_{q+1},\cdots,n_r,n_{r+1},\cdots,n_s\}$,$N_3'=\{n_r,n_{r+1},\cdots,n_s,n_{s+1},\cdots,n_n\}$,其中分割节点集$\text{node}_\xi(N,T,\rho,M,\xi)$的节点集合为:$N_\xi=\{n_p,n_{p+1},\cdots,n_q,n_r,n_{r+1},\cdots,n_s\}$.

Combiner:将节点上的Mapper输出进行规约操作,输出的<Key, Value>对通过网络送到Reducer阶段,这里的Key与Mapper输出的Key相同,即各个节点的编号值N,Value为本次迭代得到的新一轮子节点数据集.

Reducer:这个过程执行的操作跟Combiner类似,将Combiner输出的具有相同Key值的Value进行整合,得到最终的结果,并以键值对的形式输出,这里的键值对格式为:Key<节点坐标>、Value<新的节点数据集>,即依据节点的编号值次序生成节点信息文件,输出的文件可直接用于后处理模块.输出的格式应参照后处理阶段的文件格式.

由于本并行算法是在对模型文件的节点数据集进行分块的基础上实现的,各个块的程式计算理论仍与串行计算一致,因而时间复杂度为$O(n)$,空间复杂度亦为$O(n)$.

分配算法将集成到整个并行处理系统中进行验证.依据第1.2.1节的示例,后处理行为数据是程序计算输出的结果也是以数组形式存储的,每一帧的数据结构与模型文件数据结构类似.通过比较行为数据文件来验证算法的准确性.

3.2 算法验证环境及数据准备

本文的Spark集群实验平台由3台普通PC机搭建,选择其中性能较高的1台PC作为主节点(Master),运行Master节点守护进程,用于管理其他数据节点,其余2台作为从节点(Slave),运行Slave节点守护进程,作为MapReduce任务运行的工作节点.所有主机处于同一局域网中,在配置局域网IP后,Master与Slave节点之间通过安全外壳协议(SSH)方式验证.程序计算代码由python实现,而且Spark平台有针对python的接口,即pyspark,因而本实验采用Spark on YARN的运行方式.

通过选取不同大小的模型文件进行节点标识及分配,并采用不同的分区个数,同时考虑划分的分区个数对实验结果的影响.

本文通过4个空间桁架系统计算模型来验证并行算法的有效性和加速效果.前2个模型model1与model2采用桁架节点数目较小的结构,后2个模型model3与model4则参照具体的工程实例,分别为钢结构桁架桥梁与单层钢结构厂房.

其中钢结构桁架桥梁的一些基本工程参数如表1所示.

表 1　钢结构桁架桥梁结构参数

Tab.1　Parameters of steel truss bridge

主跨长度/m	节间长度/m	主桁高度/m	桥面宽度/m	桁架单元数
97	3.35	8	8	3 712

钢结构厂房的基本参数如表 2 所示.

表 2　钢结构厂房结构参数

Tab.2　Parameters of steel structure factory

开间长度/m	柱距/m	跨度/m	层高/m	屋架坡度	钢材型号
144	9	18	6	1∶3	Q235

在创建以上 2 个工程实例模型时，外力作用均模拟横向风荷载. 在本实验中，将桁架结构模型的复杂程度按照模型中的节点数目以及杆件数目作为依据进行分级，其中 model1、model2、model3 以及 model4 的节点数目和杆件数目分别如表 3 所示.

表 3　测试模型信息

Tab.3　Information of test models

项目	model1	model2	model3	model4
节点数目	64	230	530	1 110
杆件数目	128	590	1 260	2 040

综上所述，实验中用于测试的桁架结构模型按照复杂程度由低到高依次为 model1，model2，model3 以及 model4，在验证环节将通过对不同复杂程度的数据实验结果的对比，得出桁架结构模型复杂程度与 VFIFE 串行计算和并行计算效率的关系，从而归纳出 VFIFE 复杂桁架结构并行分布式处理技术的适用范围.

3.3　验证过程及结果分析

VFIFE 复杂桁架结构并行分布式处理实验包括并行计算实验与对照实验两大部分. 并行计算实验的目的是通过控制 Spark 计算平台的分区数目来控制数据集的分区数目，从而研究分区数目与模型复杂程度的关系，寻找出各个模型的最佳分区数目. 将最佳分区数目作为依据，与串行计算效率和传统有限元分析效率进行对比. 根据第 3.2 节所描述的 4 种不同复杂程度的桁架结构模型以及实验环境限制，将并行计算实验的分区数目划分为 4、8、12 这 3 种.

对照实验的验证过程针对上一节中不同复杂程度的桁架结构模型进行，可分为横向对照与纵向对照，横向对照即对照不同复杂程度的桁架结构模型在相同条件下的计算结果，目的是研究模型复杂程度对实验结果的影响. 纵向对照则针对各个模型比较，纵向对照实验主要是桁架结构 VFIFE 并行分布式处理与 VFIFE 程序串行计算效率的对照，将根据程序计算的迭代次数进行分类，进一步探寻桁架结构 VFIFE 程序计算中迭代次数与并行分布式处理效率的关系.

在并行计算实验中，同一个模型计算效率的提升以各分区数目下计算速度提升的百分比为依据，计算效率提升百分比 $\chi=\dfrac{(1/t_1-1/t_2)}{1/t_2}=\dfrac{t_2-t_1}{t_1}(t_1\leqslant t_2)$，其中 t_1 和 t_2 分别表示统一模型在不同分区数目下的计算时间.

在对照实验中，向量式有限元的串行和并行等不同计算方式的计算效率则通过比较加速比 $\zeta=t_3/t_4$ 来分析，其中 t_3 为串行计算时程序执行时间，t_4 则为相同任务量的前提下，采用并行计算算法时程序执行时间.

在所有上述实验中，最后都会将生成的行为数据输入到后处理模块中进行展示，得到更直观的结果.

对于第 3.2 节中创建的 4 个不同复杂程度的桁架结构模型进行并行计算，迭代次数设定为 10 000 次，将实验结果统计并整理后得到图 3.

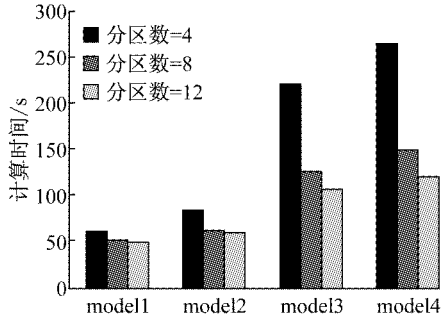

图 3　不同桁架结构模型在不同分区数目下的计算时间

Fig.3　Computing time based on different truss structure models with different partitioners

对以上结果进行分析可知，模型的复杂程度与计算所耗费的时间成正相关的关系. 随着分区数量的增加，计算效率有了一定提升，这种提升效果随着模型文件复杂程度的增加而更加明显. 在复杂模型中，分区数目增加对于计算效率提升的效果较为明显，但是在 model3 以及 model4 的实验中，分区数从 4 到 8 的计算效率提升率分别为 74.1% 以及 76.1%，而分区数从 8 到 12 的计算效率提升率分别为 17.6%以及 23.7%，提升效果相对较小. 这是因为，根据第 2.2 节对节点分配模型的定义，随着分区数目的增加，对模型文件节点数据集 node 的分配所需时间也增加，且文件结构越复杂，所耗时间越长. 这可能是由于在分配过程中需要遍历各个子节点数据集 node_i^l，并结合杆件数据集 element 来查找分割

节点,因而分区数量越多,分割节点数量越多,查找过程耗时越长.

根据上述分析,由于硬件环境限制,本文的集群能运行的最大分区约为12,当继续增加分区数量时只会增加排队等待时间的开销,而不能继续大幅度提升计算效率.因此,可以选取分区数=8作为计算各个模型的分区数目来进行对照实验.

图4为对照实验中,不同模型在不同迭代次数下并行计算相对串行计算的加速比.

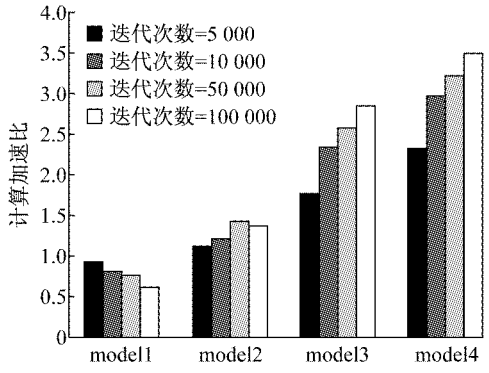

图4 不同规模桁架模型在不同迭代次数下并行相对串行的计算加速比

Fig.4 Speedup of parallel computing compared to serial computing dealing with different scales of models in different iterations

对比以上结果,分析VFIFE复杂桁架结构并行分布式处理相对串行计算的加速比,可以得到以下结论:

(1) 相对于串行计算,随着计算规模的增加,即桁架结构模型复杂程度的增加,并行计算能发挥出更好的性能.

(2) 在同一个复杂模型中,进行的迭代次数越多,并行计算的加速效果也更明显.这种加速效果在model1和model2中并不明显,但对于复杂程度较高的模型model3和model4,数据分块后并行计算带来的效益越高.

(3) model1以及model2的数据量相对较小,并行计算耗时略多于串行计算耗时,这可能是由于时间消耗在每一次迭代开始前,各个分区中的分割杆件两端节点需依据RDD中的数据同步.但总的来说,随着计算量的增加,并行计算的效率也越来越高,且与并行计算时分配的计算分区数目成一定正比关系.

4 结论与展望

(1) 本文针对节点数据集node采用的节点分配机制不会对计算结果产生影响,因为分配后的每个分区仍采用相同的计算流程.

(2) 对于数据量较大的桁架结构模型的计算,如节点数超过1 000,采用并行计算可使计算效率实现大幅提升.

(3) 在计算量较大的情况下,并行计算的效率与节点分配划分成的分区个数成一定正比关系,即分区的块越多,计算效率的提升越明显.

此后的研究应考虑划分更多块进行计算,找到最佳分块个数.完善程式计算理论,实现并行算法的通用性.

参考文献:

[1] Ting E C, Shih C, Wang Y K. Fundamentals of a vector form intrinsic finite element. Part I: basic procedure and a plane frame Element[J]. Journal of Mechanics, 2004, 20(2):113.

[2] Ting E C, Shih C, Wang Y K. Fundamentals of a vector form intrinsic finite element. Part II: plane solid elements[J]. Journal of Mechanics, 2004, 20(2):123.

[3] Ting E C, Shih C, Wang Y K. Fundamentals of a vector form intrinsic finite element. Part III: convected material frame and examples[J]. Journal of Mechanics, 2004, 20(2):133.

[4] 卢哲刚,姚谏.向量式有限元:一种新型的数值方法[J].空间结构, 2012,18(1): 85.
LU Zhegang, YAO Jian. Vector form intrinsic finite element:a new numerical method [J]. Spatial Structures, 2012, 18(1): 85.

[5] 杜庆峰,周晓玮,谢涛,等. 大规模向量式有限元行为数据压缩模型及算法[J]. 同济大学学报:自然科学版,2014, 42(11): 14.
DU Qingfeng, ZHOU Xiaowei, XIE Tao, *et al*. Massive vector form intrinsic finite element behavior data compression model and algorithm [J]. Journal of Tongji University: Natural Science, 2014, 42(11): 14.

[6] 杜庆峰,周雪非,谢涛,等. 大规模向量式有限元行为数据无损压缩模型[J]. 同济大学学报:自然科学版,2015, 43(1): 19.
DU Qingfeng, ZHOU Xuefei, XIE Tao, *et al*. Massive vector form intrinsic finite element behavior data lossless compression model[J]. Journal of Tongji University: Natural Science, 2015, 43(1): 19.

[7] 刘耀儒,周维垣,杨强. 三维有限元并行EBE方法[J].工程力学,2006,23(3): 27.
LIU Yaoru, ZHOU Weiyuan, YANG Qiang. Parallel 3D finite element analysis based on EBE method [J]. Engineering Mechanics,2006, 23(3): 27.

[8] 符伟. 云计算环境下的线性有限元方法研究[D]. 武汉:华中科技大学, 2013.
FU Wei. Study on linear finite element method in cloud computing[D]. Wuhan: Huazhong University of Science and Technology, 2013.

[9] 江雄心,万平荣. 三维有限元网格节点编号优化[J]. 工程图学学报, 2008, 29(4): 22.

JIANG Xiongxin, WAN Pingrong. Optimization of node numbering in 3D finite element[J]. Journal of Engineering Graphics, 2008,29(4):22.

[10] 倪秋斌，丁从潮，高博青，等. 基于向量式结构力学的空间桁架非线性静力分析[J]. 空间结构，2013，19(3)：33.
NI Qiubin, DING Congchao, GAO Boqing, *et al*. Nonlinear static analysis of space truss using vector mechanics of structure[J]. Spatial Structures, 2013, 19(3): 33.

[11] 荆国强，陈德伟. 一种基于代数图论的有限元模型节点排序方法[J]. 同济大学学报：自然科学版，2010，38(6)：929.
JING Guoqiang, CHEN Dewei. A finite element nodal ordering with algebraic graph theory[J]. Journal of Tongji University: Natural Science, 2010, 38(6): 929.

[12] 姜涛，王安麟，朱灯林. 有限元结点编号的综合带宽优化算法[J]. 机械设计，2005，22(11)：3.
JIANG Tao, WANG Anlin, ZHU Denglin. Synthetic bandwidth optimization algorithm of finite element node numbering[J]. Journal of Machine Design, 2005, 22(11): 3.

[13] 贾建军，彭颖红. 三种基于图论的有限元结点编号优化算法[J]. 机械科学与技术，1998，17(5)：725.
JIA Jianjun, PENG Yinghong. Three graph theory based algorithms on FEM node ordering optimization[J]. Mechanical Science & Technology, 1998, 17(5): 725.

[14] 梁彦. 基于分布式平台 Spark 和 YARN 的数据挖掘算法的并行化研究[D]. 中山：中山大学，2014.
LIANG Yan. Research on parallelization of data mining algorithm based on distributed platform Spark and YARN[D]. Zhongshan: Zhongshan University, 2014.

(上接第 1100 页)

[2] 赵波. 风洞热交换器设计研究[D]. 绵阳：中国空气动力研究与发展中心研究生部，2008.
ZHAO Bo. Investigation of heat exchanger design in wind tunnel [D]. Mianyang: The Graduate Faculty of China Aerodynamics Research and Development Center, 2008.

[3] 李启良. 风洞热交换器设计研究气动-声学风洞热交换器的数值模拟与试验研究[D]. 上海：同济大学，2007.
LI Qiliang. Numerical simulation and experimental study on heat exchanger of aero-acoustic wind tunnel[D]. Shanghai: Tongji University, 2007.

[4] Tahseen A T, Ishak M, Rahman M M. An overview on thermal and fluid flow characteristics in a plain plate finned and un-finned tube banks heat exchanger [J]. Renewable and Sustainable Energy Reviews, 2015, 43: 363.

[5] 赵兰萍，杨志刚. 管束排列方式对矩形翅片椭圆管束性能的影响[J]. 同济大学学报：自然科学版，2016，44(2)：298.
ZHAO Lanping, YANG Zhigang. Effect of tube arrangement on the performance of rectangular finned elliptical tube bundles [J]. Journal of Tongji University: Natural Science, 2016, 44(2): 298.

[6] 赵兰萍，杨志刚. 管间距对矩形翅片椭圆管换热管束性能的影响[J]. 同济大学学报：自然科学版，2016，44(1)：150.
ZHAO Lanping, YANG Zhigang. Effect of tube pitches on the performance of rectangular finned elliptical tube heat exchanger[J]. Journal of Tongji University: Natural Science, 2016, 44(1): 150.

[7] 刘宝兴，蔡祖恢. 空气横掠椭圆矩形翅片管束的放热和阻力性能的试验研究[J]. 工程热物理学报，1982，3(4)：365.
LIU Baoxing, CAI Zhuhui. Heat transfer and draft loss performance of air flowing across staggered banks of oval tubes fitted with rectangular fins [J]. Journal of Engineering Thermophysics, 1982, 3(4): 365.

[8] 杨立军，贾思宁，卜永东，等. 电站间冷系统空冷散热器翅片管束流动传热性能的数值研究[J]. 中国电机工程学报，2012，32(32)：50.
YANG Lijun, JIA Sining, BU Yongdong, *et al*. Numerical study on flow and heat transfer characteristics of finned tube bundles for air-cooled heat exchangers of indirect dry cooling systems in power plants[J]. CIESC Journal, 2012, 32(32): 50.

[9] 杨立军，张凯峰，杜小泽，等. 空冷凝汽器椭圆翅片椭圆管束外空气的流动与传热特性[J]. 动力工程，2008，28(6)：911.
YANG Lijun, ZHANG Kaifeng, DU Xiaoze, *et al*. Flow and heat transfer characteristics of cooling air outside elliptical tube bundles fixed with elliptical fin in air-cooled condenser [J]. Journal of Power Engineering, 2008, 28(6): 911.

[10] 马义伟. 空冷器设计与应用[M]. 哈尔滨：哈尔滨工业大学出版社，1998.
MA Yiwei. Design and application of air condenser [M]. Harbin: Harbin Institute of Technology Press, 1998.

[11] Stephan K. Heat and mass transfer [M]. Berlin: Springer-Verlag, 2006.

[12] 张来，杜小泽，杨立军，等. 开孔矩形翅片椭圆管流动与换热特性的数值研究[J]. 工程热物理学报，2006，27(6)：990.
ZHAO Lai, DU Xiaoze, YANG Lijun, *et al*. The flow and heat transfer characteristics of numerical investigation of rectangular-fin elliptical-tube with interrupted holes [J]. Journal of Engineering Thermophysics, 2006, 27(6): 990.