

基于模糊聚类分析的构件并行技术研究

杜 静¹⁾ 敖富江²⁾ 杨学军¹⁾ 杨灿群¹⁾

¹⁾(国防科技大学计算机学院 长沙 410073)

²⁾(国防科技大学机电工程与自动化学院 长沙 410073)

摘 要 提出了一种新的面向科学计算的构件技术——基于模糊聚类分析的构件并行技术,旨在提高构件间的并行度和数据局部性,避免通信瓶颈.该技术分为两个阶段:域划分和子构件组合.其中域划分利用了编译时的数据依赖分析技术,然后考虑访问步长的影响,利用不定方程,提出了区间重叠度的概念.基于此利用所设计的基于区间重叠度的模糊聚类算法实现子构件分类组合,并给出了算法的形式化描述.实验结果表明,通过该算法的编译时优化,构件程序能够获得良好的数据局部性、适中的粒度以及高度的并行性,算法具有很好的可扩展性.

关键词 通信瓶颈;并行度;数据局部性;区间重叠度;模糊聚类

中图法分类号 TP311

Research on Component Parallel Technology Based on Fuzzy Clustering Analysis

DU Jing¹⁾ AO Fu-Jiang²⁾ YANG Xue-Jun¹⁾ YANG Can-Qun¹⁾

¹⁾(School of Computer, National University of Defense Technology, Changsha 410073)

²⁾(School of Mechatronics and Automation, National University of Defense Technology, Changsha 410073)

Abstract This paper proposes a new scientific computing-oriented component technology—Component Parallel Technology Based on Fuzzy Clustering Analysis, aiming at improving parallelism and data locality among components, and avoiding communication bottleneck. The technology is composed of two parts: Domain partition and sub-component combination. Domain partition uses data dependence analysis technique during compile time. Then considering the effect of access stride, the concept of interval overlap degree is proposed by using indefinite equation. Based on this, it implements the classification and combination of sub-components by using fuzzy clustering algorithm for interval overlap degree designed by the authors, and presents the formal description of the algorithm. The experimental results show that the algorithm is efficient and scalable for scientific component programs in terms of fine data locality, moderate granularity and high parallelism.

Keywords communication bottleneck; parallelism; data locality; interval overlap degree; fuzzy clustering

1 引 言

复杂性呈几何级数增长^[1].将构件技术^[2-3]引入科学计算领域能够充分利用高效昂贵的计算资源,成为目前高性能计算领域新的研究热点^[4-5].

随着科学计算问题规模增大,传统并行软件的

面向科学计算的构件技术将并行软件封装成构

收稿日期:2006-03-13;最终修改稿收到日期:2007-04-14. 本课题得到国家自然科学基金(60621003,60633050)资助. 杜 静,女,1979年生,博士研究生,主要研究方向为高性能计算、并行算法和编程. E-mail: jdstarry@yahoo.com.cn. 敖富江,男,1975年生,博士研究生,主要研究方向为复杂大系统的建模与仿真、数据挖掘. 杨学军,男,1963年生,教授,博士生导师,主要研究领域为高性能计算、并行体系结构、高性能编译和操作系统. 杨灿群,男,1968年生,副研究员,主要研究领域为高性能编译与并行编程.

件^[6],尽可能多地满足应用领域的特定需求,为科学计算并行程序开发人员提供一个易使用、易复用和易扩展的新型并行开发环境,从而缩短开发时间,减少系统的复杂度,提高系统的可靠性,可扩展性以及可维护性.由于科学计算领域通常涉及到的数据量巨大,因此其构件并行模型主要以数据并行为主.现有的并行系统大多支持构件内的数据并行和构件间的控制并行,但是构件内的数据并行需要在各并行构件中对数据集进行分解与合成,不仅增加了开销,而且使构件之间存在通信瓶颈.为了提高数据处理的效率,提高系统并行度和数据局部性,避免通信瓶颈,在文献[7]中,我们提出了编译指导的构件并行方法.然而该方法没有考虑循环步长对数据访问模式的影响,系统性能优化受限.基于文献[7],本文提出了基于模糊聚类分析的构件并行技术.该技术主要针对循环内的数组,首先根据编译时的数据依赖分析将粗粒度构件划分为子构件,然后考虑访问步长的影响,利用不定方程,提出了区间重叠度的概念,准确反映了区间的数据相关性.基于此,使用我们所设计的基于区间重叠度的模糊聚类算法将子构件分类,重新组合成粒度适中且数据局部性好的并行构件,从而提高系统的性能.实验结果表明,该算法能够开发科学构件系统的性能潜力,具有很好的可扩展性.

2 基于模糊聚类分析的构件并行模型

一个构件的本质是一个计算单元,主要由两种值组成:基本值和对象^[8-9].基本值是一个数学抽象,诸如整数、集合以及堆栈等,而一个构件中的对象则是一个值的容器,它可以是一个变量(包括简单变量和数组)或者一个子构件.对于并行构件,不同构件对同一数据的频繁访问限制了数据并行.为了准确判断构件之间的数据相关性,并行构件的说明应当能够形式化地描述构件对数据的访问情况,包括全局对象的定义,每个操作的语法信息以及每个构件在执行过程中访问的数据名等,尤其当访问对象是数组元素时,枚举每个元素必然很繁琐,因此对于同一数组元素,我们应当描述可能被访问的数组的下标范围.

一个应用程序是由多个功能独立的构件通过数据流通道互联构成,每个构件带有若干输入和输出,构件的功能是将其输入的数据进行变换后形成数据流送给与之相连的其他构件.与传统的控

制流思想相比,数据流的思想非常适用于并行和模块级编程^[10].基于此,我们设计了基于模糊聚类分析的构件并行模型,旨在提高构件间的并行性和数据局部性,避免通信瓶颈,在可利用的资源基础上,最大化流量的同时最小化执行时间.该模型通过编译时分析,分为两个阶段:域划分和子构件组合,简述如下.

- (1)域划分.将计算规模庞大的构件进行适当划分,目的是尽量开拓构件内并行执行的机会.
- (2)子构件组合.按照构件间的数据相关性及访问局部性,考察域划分的结果.运用我们所设计的基于区间重叠度的模糊聚类的方法对子构件进行分类组合,从而提高构件间的并行性和数据局部性,并避免通信瓶颈.

图1举例说明了基于模糊聚类分析的构件并行模型的执行过程.图中矩形表示构件, $S_{i,j}(x)$ 表示*i*语句和*j*语句都引用变量*x*.源程序包含两个构件(构件X和构件Y),分别具有4条语句和1条语句.经过编译器分析可知构件X的1号语句和2号语句具有对变量*a*的数据依赖,首先对源程序进行域划分以拓展构件X的并行度,构件X被分解为数据相关性小的3个子构件,具有数据依赖的1号和2号语句被聚集于同一子构件.域划分后形成可并行执行的4个子构件.其次1号和2号语句所在的子构件与5号语句所在的子构件具有对变量*a*的数据重用,运用基于区间数据模糊聚类方法组合子构件

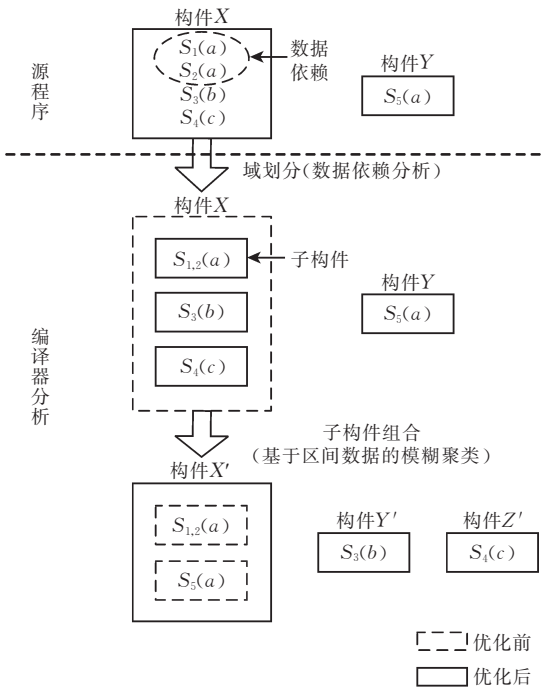


图1 基于模糊聚类分析的构件并行模型

形成计算密集的构件 X' ，最终重组为可并行执行的 3 个构件 X' 、 Y' 和 Z' ，构件内部具有良好的数据局部性。

2.1 域划分

域划分就是将构件所访问的数据划分成一些小的数据子集，形成多个可并行的子构件，以充分揭示并行执行的机会。而数据依赖关系是影响程序并行化的关键因素，因此域划分的依据采用数据依赖分析技术。

首先，嵌套循环中的标量和数组是科学计算构件访问的主要数据，而标量可以作为数组的退化情况，且不同数组的元素决不会引用相同的存储单元，因此域划分所需的数据依赖分析是判断循环中两个下标索引向量的同名数组元素在给定条件下是否表示同一存储单元。在实际程序中，绝大多数数组元素引用的下标均是循环索引向量的线性表达式^[11]，因此我们主要分析线性下标数组元素引用导致的依赖关系。

为了便于编译器分析，根据文献[11]建立数组依赖关系的数学模型，从而获得嵌套循环中语句之间存在依赖关系的必要条件。

考虑嵌套循环 L ，设 S 和 T 为其循环体中的两个语句， A 为一个 n 维数组， x 和 y 分别为 S 和 T 对数组 A 的引用，且至少有一个是其所在语句的输出变量。设 $x = A(f_1(I), f_2(I), \dots, f_n(I))$ ， $y = A(g_1(I), g_2(I), \dots, g_n(I))$ ， $f_k(I)$ 和 $g_k(I) (1 \leq k \leq n)$ 均是索引向量 $I = (I_1, I_2, \dots, I_m)$ 的线性整数函数。如果这两个变量导致 S 和 T 之间的依赖关系（表示为 δ ），则方程组 $f_k(I) - g_k(I) = 0$ 存在满足约束条件： $p_r \leq i_r \leq q_r, p_r \leq j_r \leq q_r (0 \leq r \leq m)$ 的整数解 (i, j) ，其中 $i = (i_1, i_2, \dots, i_m)$ ， $j = (j_1, j_2, \dots, j_m)$ 。且

- (1) 若 $i < j$ ，则 $S\delta T$ ；
- (2) 若 $i > j$ ，则 $T\delta S$ ；
- (3) 若 $i = j$ 且 $S < T$ ，则 $S\delta T$ 。

编译器根据以上数学模型分析构件内部数组之间的数据依赖关系，将数据集分解为数据相关性小的多个子集，形成多个可并行执行的子构件，提高构件内的数据并行性。

2.2 子构件组合

子构件组合的目的是通过合并域划分产生的子构件来减少任务数。由于子构件的组合方式并不唯一，因此根据子构件之间的数据重用度及数据局部性，按照模糊聚类的方法，将子构件分类组合，形成

新的并行构件。新构件具有良好的数据重用性、局部性以及适中的粒度，从而提高构件间的并行性，避免通信瓶颈。

目前的模糊聚类分析方法主要用于将属性值是单个数字的对象分类^[12-13]，而我们所考察的子构件中的数组是在一定范围内变化的。通过分析不同子构件对相同数组的访问区间，考虑访问步长的影响，得出区间之间的访问重叠度。重叠度越大，说明子构件之间关于该数组的数据相关性越大，即经常访问相同数据。然后对子构件进行聚类分析，将数据相关性大且数据局部性高的子构件分类组合，从而使新构件访问外部数据的频率减少，同时又使原本分散的数据访问集中起来。因此针对子构件分类问题，我们提出了一种基于区间重叠度的聚类算法。算法的步骤如下。

2.2.1 数据的获取、表示及其标准化

假定域划分之后，系统中有 n 个待分类的子构件，共有 m 个数组且约定顺序，记为 F_m 。建立模型：待分类对象集合为 $X = \{X_1, X_2, \dots, X_n\}$ 。集合中的每个元素具有 m 个属性，对应于 m 个数组。设第 i 个对象 X_i 的第 $j (j = 1, 2, \dots, m)$ 个属性值为 x_{ij} 。则 X_i 就可以用这 m 个属性的取值来描述，记

$$X_i = (x_{i1}, x_{i2}, \dots, x_{im}) (i = 1, 2, \dots, n),$$

这里 X 代表所有子构件的集合，子构件 X_i 中的 m 个元素分别代表 m 个有序数组被子构件访问的范围以及访问步长。与传统模糊聚类方法不同的是，对象的属性值 x_{ij} 不使用确定的数值，而直接用一个集合表示，集合元素为访问闭区间 $[a_{ij}, b_{ij}]$ 和访问步长 t_{ij} 的组合，即每个对象的值为一个以集合为分量的一个向量，可以记为

$$x_{ij} = \{[a_{ij}, b_{ij}], t_{ij}\}$$
$$X_i = \{\{[a_{i1}, b_{i1}], t_{i1}\}, \{[a_{i2}, b_{i2}], t_{i2}\}, \dots, \{[a_{im}, b_{im}], t_{im}\}\} (i = 1, 2, \dots, n),$$

$[a_{ij}, b_{ij}]$ 代表子构件 X_i 访问数组 F_j 的下标范围。区间的端点表示其下标变化的起始点，区间的长度表示其变化的范围。数据表示如图 2 所示。

访问区间 子构件	数 组	分类对象的属性			
		F_1	F_2	\dots	F_m
待 分 类 对 象	X_1	$\{[a_{11}, b_{11}], t_{11}\}$	$\{[a_{12}, b_{12}], t_{12}\}$	\dots	$\{[a_{1m}, b_{1m}], t_{1m}\}$
	X_2	$\{[a_{21}, b_{21}], t_{21}\}$	$\{[a_{22}, b_{22}], t_{22}\}$	\dots	$\{[a_{2m}, b_{2m}], t_{2m}\}$
	\vdots	\vdots	\vdots	\dots	\vdots
	X_n	$\{[a_{n1}, b_{n1}], t_{n1}\}$	$\{[a_{n2}, b_{n2}], t_{n2}\}$	\dots	$\{[a_{nm}, b_{nm}], t_{nm}\}$

图 2 基于区间重叠度的模糊聚类算法的数据表示

以不同步长访问同一数组的两个区间时，为了

更加准确地表示区间之间的数据相关性,下文提出了区间重叠度的概念,即两个区间被重叠访问的数据的数目.

设子构件 X_i 以步长为 t 访问数组 F_j , 访问的下标区间为 $[a, b]$, 则区间内被访问数据的下标为 $a + xt \in [a, b]$, $t, x \in N$, 若步长为负, 进行区间翻转变换即可. 若子构件 X_i 和 X_j 分别以步长 t_1 和 t_2 访问数组 F_m 的下标区间为 $[a, b]$ 和 $[c, d]$, 我们希望得到区间 $[a, b]$ 和 $[c, d]$ 中被重叠访问的数据数目, $t_1, t_2 \in N$. 以此定义区间的重叠度. 算法如下(算法所涉及变量的含义如图 3 所示).

1. 求区间 $[a, \infty]$ 和 $[c, \infty]$ 中被重叠访问的所有相同数据 x 和 y .

求解不定方程 $a + xt_1 = c + yt_2$, $x, y \in N^{[14]}$.

若 $(t_1, -t_2) | (c - a)$, 则该不定方程有解, 其全部解为

$$\begin{cases} x = x_0 + \frac{t_2}{(t_1, -t_2)}k, & k \in N \\ y = y_0 + \frac{t_1}{(t_1, -t_2)}k, & k \in N \end{cases} \quad (1)$$

$$\begin{cases} x = x_0 + \frac{t_2}{(t_1, -t_2)}k, & k \in N \\ y = y_0 + \frac{t_1}{(t_1, -t_2)}k, & k \in N \end{cases} \quad (2)$$

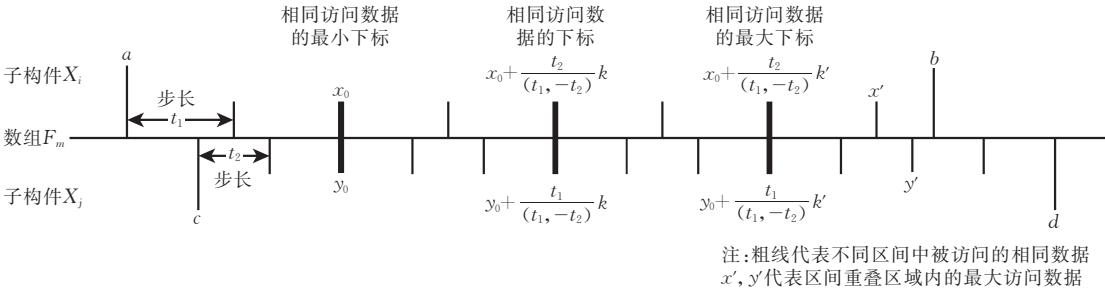


图 3 两个子构件以不同步长对同一数组的访问情况

运用反证法易证 $[a, x']$ 中被重叠访问的数据即为 $[c, y']$ 中被重叠访问的数据, 因此不等式(5)和不等式(6)同解(证明略). 取不等式(5)或(6)的解的最大值 $k', k' + 1$ 则为区间 $[a, x']$ 和 $[c, y']$ 中被重叠访问的数据数目. 由于 x' 和 y' 是 $[a, b]$ 和 $[c, d]$ 重叠区域的最大访问数据, 则易见区间 $[a, x']$ 和 $[c, y']$ 中被重叠访问的数据数目等同于区间 $[a, b]$ 和 $[c, d]$ 中被重叠访问的数据数目.

$$k' = \left\lfloor \left[\frac{\min(b, d) - a}{t_1} - x_0 \right] \frac{(t_1, -t_2)}{t_2} \right\rfloor$$

或

$$k' = \left\lfloor \left[\frac{\min(b, d) - c}{t_2} - y_0 \right] \frac{(t_1, -t_2)}{t_1} \right\rfloor.$$

为了便于对属性状态值的表示加以讨论, 先给出有关区间的某些约定:

本文中 Z 表示整数集, $[Z] = \{[a, b]: a < b, a, b \in Z\}$.

定义 1. 设 $[a, b], [c, d] \in [Z], [a, b]$ 中被访

$(t_1, -t_2)$ 代表 $t_1, -t_2$ 的最大公因数, x_0 和 y_0 是该方程最小的一组解, 即代表两个区间首次重叠访问的数据下标, 利用辗转相除法可以求得.

2. 求区间 $[a, b]$ 和 $[c, d]$ 重叠区域内的最大访问数据 x' 和 y' . 即求以下不等式方程组的最大整数解 x' 和 y' :

$$\begin{cases} a + xt_1 \leq \min(b, d) \\ c + yt_2 \leq \min(b, d) \end{cases},$$

解得

$$x' = \left\lfloor \frac{\min(b, d) - a}{t_1} \right\rfloor \quad (3)$$

$$y' = \left\lfloor \frac{\min(b, d) - c}{t_2} \right\rfloor \quad (4)$$

3. 求区间 $[a, x']$ 和 $[c, y']$ 中被重叠访问的所有数据的数目. 即求解 $x \leq x'$ 或 $y \leq y'$ 的所有 x 或 y 的数目. 将步 1 中式(1), (2)的 x, y 值和步 2 中式(3), (4)的 x' 和 y' 值代入, 得到不等式方程

$$x_0 + \frac{t_2}{(t_1, -t_2)}k \leq \left\lfloor \frac{\min(b, d) - a}{t_1} \right\rfloor \quad (5)$$

$$y_0 + \frac{t_1}{(t_1, -t_2)}k \leq \left\lfloor \frac{\min(b, d) - c}{t_2} \right\rfloor \quad (6)$$

问的最大数据为 x' ,

(1) 若 $c \geq a$, 称 $[c, d]$ 大于 $[a, b]$, 记作 $[c, d] \geq [a, b]$, 或 $[a, b] \leq [c, d]$;

(2) 若 $c > x'$, 称 $[c, d]$ 绝对大于 $[a, b]$, 记作 $[c, d] > [a, b]$, 或 $[a, b] < [c, d]$.

定义 2. 设 $[a, b], [c, d] \in [Z], t_1, t_2 \in N$ 代表访问各自区间的步长. 不失一般性, 取 $[c, d] \geq [a, b]$. 若 $\neg([c, d] > [a, b])$, 称 $[a, b], [c, d]$ 相互重叠. 其重叠度 O 定义为

$$O(\{[a, b], t_1\}, \{[c, d], t_2\}) = \begin{cases} 0, & [c, d] > [a, b] \\ k' + 1, & \text{否则} \end{cases}.$$

2.2.2 建立相似矩阵, 确定各个对象间的相似度

在得出了各个子构件对应于每个数组的访问区间后, 考虑将各个子构件间的相似度组成一个 n 阶相似矩阵, 它是 X 上的模糊相似关系. 由于本文所讨论的属性值是用区间和步长表示的, 因此传统相似度的求法不再适用, 必须探讨一种新的求相似度

的方法.

设子构件 $X_i, X_j \in X$, 其属性集合分别为

$$X_i = \{ \{ [a_{i1}, b_{i1}], t_{i1} \}, \{ [a_{i2}, b_{i2}], t_{i2} \}, \dots, \{ [a_{im}, b_{im}], t_{im} \} \} \quad (i=1, 2, \dots, n),$$

$$X_j = \{ \{ [a_{j1}, b_{j1}], t_{j1} \}, \{ [a_{j2}, b_{j2}], t_{j2} \}, \dots, \{ [a_{jm}, b_{jm}], t_{jm} \} \} \quad (j=1, 2, \dots, n),$$

即 X_i 和 X_j 对数组 F_k 的访问下标区间为 $[a_{ik}, b_{ik}]$ 和 $[a_{jk}, b_{jk}]$, 访问步长分别为 t_{ik} 和 t_{jk} , $k=1, 2, \dots, m$. 根据定义 2, 任意区间 $[a_{ik}, b_{ik}], [a_{jk}, b_{jk}]$ 的重叠度为 $O(\{ [a_{ik}, b_{ik}], t_{ik} \}, \{ [a_{jk}, b_{jk}], t_{jk} \})$. 并且考虑不同数组的访问频率不同, 对于子构件分类所占的重要程度不同, 我们可以通过编译时分析确定访问频率所占的权重, 提高分类结果的数据局部性. 两个对象 X_i, X_j 间的相似度 S_{ij} 定义如下.

定义 3. 设各个数组的访问频率分别为 $\{\omega_1, \omega_2, \dots, \omega_m\}$, 子构件 X_i, X_j 间的相似度定义为

$$S_{ij} = N \left(\sum_{k=1}^m \omega_k \cdot O(\{ [a_{ik}, b_{ik}], t_{ik} \}, \{ [a_{jk}, b_{jk}], t_{jk} \}) \right),$$

其中, \cdot 为乘法运算, N 为矩阵的归一化函数. 则易见

- (1) $S_{ij} \in [0, 1]$;
- (2) $S_{ij} = 1$ (自反性);
- (3) $S_{ij} = S_{ji}$ (对称性).

因此以 S_{ij} 为元素的矩阵为模糊相似矩阵^[12].

2.2.3 得出聚类结果

得出了模糊相似矩阵 S 后, 求其传递闭包, 得到模糊等价矩阵 $t(S)$ ^[12-13]. 根据不同的置信水平 α , 求 $t(S)$ 的 α 截关系 $e(S)_\alpha$, 得到普通的分类关系. 当 $\alpha=1$ 时, 每个对象自成一类, 随着 α 值的降低, 由细到粗逐渐归并, 最后得到动态聚类谱系图. 注意灵活调整分类的置信水平 α , 控制分类结果的粒度. 粒度太大则仍然存在通信瓶颈, 大量细粒度的任务则可能增加通信代价.

通过使用基于区间重叠度的模糊聚类方法, 编译时将域划分后的所有子构件按照数组重叠度和数据局部性分类, 然后根据聚类结果进行构件重组. 新构件具有良好的数据局部性、适中的粒度, 构件之间的数据相关性小, 从而提高了系统的并行度, 避免了通信瓶颈.

2.3 算法的形式化描述

为了便于表示, 定义以下函数.

判断数据依赖关系函数: $DEP(x, y)$. 数据子集 x 和 y 存在数据依赖关系.

构件划分函数: $Partition(x, y)$. 将数据子集 x

和 y 划分为两个子构件.

计算区间重叠度函数: $Overlap(x_i, x_j, f_k)$. 计算子构件 x_i, x_j 访问数组 f_k 时的区间重叠度.

矩阵归一化函数: $Normalize(S)$. S 矩阵归一化, 返回相似矩阵.

模糊聚类函数: $FuzzyCluster(S', \alpha)$. 将相似矩阵 S' 以置信水平 α 进行模糊聚类, 返回分类数组.

构件重组函数: $Add(x_i, Component_j)$. 将子构件 x_i 加入构件 $Component_j$.

n 代表子构件数目, m 代表数组数目. 该算法的形式化描述如下:

```

for each Component do
  if (  $\forall x_1 x_2 \in \text{Component} \rightarrow \neg DEP(x_1, x_2)$  )
    // 构件中的两个数据子集没有依赖关系
    Partition( $x_1, x_2$ );

for (  $i=1; i < n+1; i++$  )
  for (  $j=i+1; j < n+1; j++$  ) {
     $S_{ij} = 0$ ;
    for (  $k=1; k < m+1; k++$  ) {
       $O = Overlap(x_i, x_j, f_k)$ ; // 计算区间重叠度
       $S_{ij} = S_{ij} + \omega_k O$ ; // 计算构件相似关系
    }
  }

 $S' = Normalize(S)$ ; // 归一化, 得到相似矩阵
Cluster[] = FuzzyCluster( $S, \alpha$ );
// 对相似矩阵进行模糊聚类, 得到分类结果
// Cluster[], 调整置信水平  $\alpha$ , 控制分类规模

for (  $i=1; i < n+1; i++$  )
  // 根据分类结果, 子构件重新组合
  for (  $j=1; j++$  )
    if (  $(x_i \in Cluster_j \wedge \neg (x_i \in Component_j))$  ) {
      Add( $x_i, Component_j$ );
      Break;
    }

```

3 实验结果及性能分析

3.1 实验结果

为了验证本文提出的算法对并行构件系统性能的影响, 我们选择一组以数组访问为主的科学计算程序进行测试. 首先将测试程序转化为构件程序, 在 SMP 和 CC-NUMA 平台上分别测试了不考虑访问步长、考虑访问步长以及不进行优化 3 种不同情况下的优化效果.

以 ftA 和 vpenta 为例, ftA 取自 NAS 基准程

序,是一个三维 fft PDE 问题.程序前面几个计算段对数组的访问是连续的规则访问,后面两个计算段则不是连续访问,程序中的数组规模为 $128 \times 128 \times 64$. Vpenta 是取自 Spec92/NASA 基准程序测试包中的测试程序,程序各个计算段的局部性均较好,程序中的数组规模为 920×920 .

测试结果表明,通过控制基于模糊聚类的构件并行算法的聚类因子,优化后的构件粒度趋于均匀,避免因构件间粒度不均引起的通信瓶颈.如图 4 所示,其中相对粒度指构件粒度与系统中最小构件粒

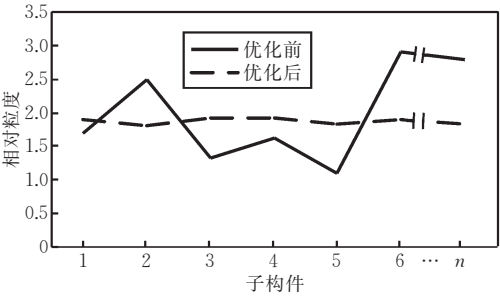
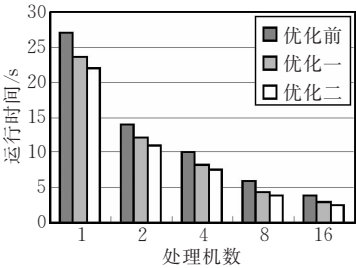
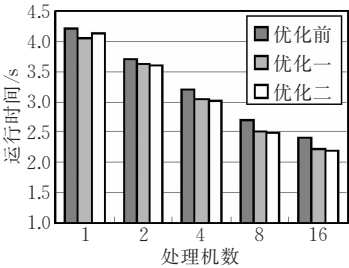


图 4 算法优化对子构件粒度的影响

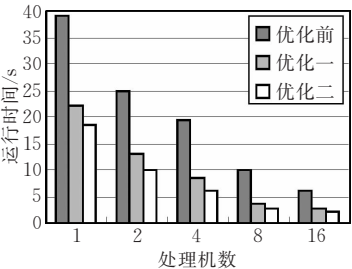


(a) ftA在SMP上的运行情况

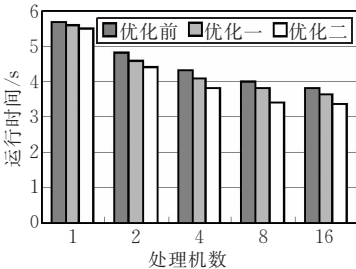


(b) vpenta在SMP上的运行情况

图 5 SMP 体系结构上的实验结果



(a) ftA在CC-NUMA上的运行情况



(b) vpenta在CC-NUMA上的运行情况

图 6 CC-NUMA 体系结构上的实验结果

3.2 性能分析

基于模糊聚类分析的构件并行技术旨在提高构件间的数据并行性、数据局部性、避免粗粒度构件产生的通信瓶颈,存在以下优势:

- (1) 通过编译时的域划分,充分开发构件内部的数据并行性.
- (2) 考虑访问步长的影响,形成不同子构件对

度的比值.

我们分别在两种平台上测试了上述各题优化前后的效率,结果如下:

(1) 图 5 是在 SMP 体系结构上的测试结果.由于 SMP 系统不存在远程本地存储访问开销的差别,因此没有数据局部性的影响.两个测试程序优化效果不明显.

(2) 图 6 是在 CC-NUMA 体系结构上的测试结果.由于 CC-NUMA 系统远程数据的访问开销远远大于访问本地数据的开销,因此良好的数据局部性、适中的粒度以及高度的并行性使得测试程序在该体系结构上的优化效果明显.

测试结果表明,我们的优化方法对类似 ftA 存储访问特征的程序优化效果好.而通过对 vpenta 程序的测试,表明模糊聚类指导的构件并行方法对具有规则访问模式的程序优化效果不明显.并且随着处理器数目的递增,其优化效果的差距越来越明显,并行编译获得了良好的并行效果,该算法具有良好的可扩展性.

相同数组的区间访问重叠度,准确反映了构件之间针对特定数组的数据相关性.

(3) 通过分析区间访问重叠度,将具有较强数据相关性的子构件分为一类,提高了构件之间的并行性.

(4) 将数组访问频率作为分类权值,保证访问同一数组局部性高的构件成为一类,提高系统的数

据局部性。

(5) 模糊聚类是实现不确定分类的有效方法, 分类的结果较为稳定, 准确性较高; 通过灵活控制聚类算法的置信因子, 保证新的并行构件的规模, 避免粗粒度构件的通信瓶颈; 防止产生细粒度的构件, 减小通信代价。

4 结束语

近年来, 随着应用需求的驱动, 针对高性能科学计算的构件技术成为了研究热点。为了提高构件间的并行性, 增强构件的数据局部性, 避免通信瓶颈问题, 本文提出了基于模糊聚类分析的构件并行技术。该技术准确反映了访问步长对构件间数据相关性的影响, 通过采用我们所设计的基于区间重叠度的模糊聚类算法, 将构件重新划分组合, 保证了系统运行时的效率。

目前这种技术还需要考虑许多问题, 包括设计更准确高效的域划分算法, 以及考虑通信因素对分类结果的影响等。这也是今后我们将进一步研究和解决的内容。

致 谢 在此, 我们向对本文的工作给予支持和建议的同行, 尤其是国防科技大学杨学军教授领导的讨论班上的同学和老师表示感谢!

参 考 文 献

- [1] Ray J et al. Performance measurement and modeling of component applications in a high performance computing environment: A case study//Proceedings of the 18th International Parallel and Distributed Computing Symposium. Santa Fe, NM, 2004; 95b
- [2] Object Management Group. CORBA component model specification (3.0 Edition). OMG Document, 2002
- [3] Gannon D et al. Programming the grid: Distributed software components, P2P and grid Web services for scientific applications. In Special Issue on Grid Computing, Journal of Cluster

Computing, 2002, 5(3): 325-336

- [4] Armstrong R, Gannon D et al. Toward a common component architecture for high performance scientific computing//Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computation. Redondo Beach, CA, 1999; 115-124
- [5] Lefantzi S et al. Using the common component architecture to design high performance scientific simulation codes//Proceedings of the International Parallel and Distributed Processing Symposium. Nice, France, 2003; 52a
- [6] Parker S G. A component-based architecture for parallel multi-physics PDE simulation//Proceedings of the International Conference on Computational Science. LNCS 2331. Springer-Verlag, 2002; 719-734
- [7] Du Jing et al. Compilation guided component parallel technology//Proceedings of the DPCS2005. Shanghai, China, 2005; 11-13(in Chinese)
(杜静等. 编译指导的构件并行技术//2005 年全国开放式分布与并行计算学术会议. 上海, 中国, 2005; 11-13)
- [8] Richard B et al. A soft engineering experiment in software component generation//Proceedings of the ICSE. Berlin, Germany, 1996; 542-552
- [9] Yang Meng Tan. Interface language for supporting programming style. ACM SIGPLAN Notices, 1994, 29(8): 74-83
- [10] Rapps S, Weyuker E J. Selecting software test data using data flow information. IEEE Transactions on Software Engineering, 1985, 11(4): 367-375
- [11] Sheng Zhi-Yu, Hu Zi-Ang et al. Methods of Parallel Compilation. Beijing: National Defence Industry Press, 2000 (in Chinese)
(沈志宇, 胡子昂等. 并行编译方法. 北京: 国防工业出版社, 2000)
- [12] Liu Pu-Yin, Wu Meng-Da. Fuzzy Theory and Application. Changsha: National University of Defense Technology Press, 1998(in Chinese)
(刘普寅, 吴孟达. 模糊理论及其应用. 长沙: 国防科技大学出版社, 1998)
- [13] Timm H, Kruse R. A modification to improve possibilistic fuzzy cluster analysis//Proceedings of the 2002 IEEE International Conference on Fuzzy Systems. Honolulu: IEEE, 2002, 2: 1460-1465
- [14] Li Chao, Xie Rui-Qiang. Algebra Foundation. Changsha: National University of Defense Technology Press, 2000 (in Chinese)
(李超, 谢端强. 代数学基础. 长沙: 国防科技大学出版社, 2000)



DU Jing, born in 1979, Ph. D. candidate. Her research interests include high performance computing, parallel algorithms and programming.

search interests include complex system modeling and simulation and data mining technique.

YANG Xue-Jun, born in 1963, professor, Ph. D. supervisor. His research interests include high performance computing, parallel computer architecture, high performance compiler and operating system.

YANG Can-Qun, born in 1968, associate professor. His research interests include high performance compiler and parallel programming.

AO Fu-Jiang, born in 1975, Ph. D. candidate. His re-

Background

With the advancement of high performance computing, the complexity of scientific computing software has grown tremendously. It is significant to use component technology in high performance scientific computing for the needs of large-scale, complex, high-performance scientific software. The key technique of using scientific component is how to execute the components in parallel more efficiently. The existing systems just reorganize the data set within components at run-time to perform data parallelism, and thereby increase the run-time overhead, restrict data parallelism among components and cause the communication bottleneck between components. To address the problems, the authors propose a new compile-time scientific computing-oriented component technology—Component Parallel Technology Based on Fuzzy Clustering Analysis, aiming at improving parallelism and data

locality among components, and avoiding communication bottleneck. This paper presents the details of the new technique, including domain partition via data dependence analysis, classification and combination of sub-components by using fuzzy clustering algorithm for interval overlap degree designed by the authors and the corresponding formal algorithm description. The experimental results show the efficiency of the new scientific computing-oriented component technology.

The work of this paper was supported by the Natural Science Foundation of China under grant Nos. 60621003 and 60633050, in which the efforts in this paper focus on solve the problems of programming complexity and parallel efficiency. The research group has tested plentiful experiments and gotten a lot of good efforts in scientific computing components.