

# 一种改进的并行计算图划分模型

马永刚, 谭国真, 杨际祥, 潘 东

(大连理工大学 计算机科学与技术学院, 辽宁 大连 116024)

E-mail: mayonggang2006@hotmail.com

**摘要:** 图划分成功地应用在许多领域, 但应用于并行计算时, 使用边割度量通信量, 其主要缺点是不能准确代表通信量, 而且图划分模型没有考虑通信延迟和通信额外开销的分布对并行性能的影响. 提出了改进的图划分模型, 该模型将影响并行性能的多个要素 (通信延迟、最大的局部通信额外开销和整体通信额外开销) 整合到一个统一的代价函数, 不仅克服了图划分模型中边割度量的一些缺点, 而且可以通过调整加权参数, 处理不同的优化目标和强调不同因素对并行性能的影响.

**关键词:** 图划分; 并行计算; 负载均衡; 通信额外开销

中图分类号: TP301

文献标识码: A

文章编号: 1000-1220(2011)03-0416-05

## Improved Graph Partitioning Model for Parallel Computing

MA Yong-gang, TAN Guo-zhen, YANG Ji-xiang, PAN Dong

(School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China)

**Abstract:** Graph partitioning has been successfully applied in many areas. However, the graph partitioning is employed in parallel computing using the edge cuts as the communication volume; the main shortcoming is that edge cuts are not entirely the same as the communication volume. Moreover, the graph partitioning model does not consider the effects of communication latency and communication overhead distribution on the parallel performance. This paper thus proposes an improved graph partitioning model for parallel computing, which incorporates several factors (communication latency, maximum local communication overhead and total communication overhead) into a unified cost function. The model not only can overcome shortcomings of the edge cut metric, but also can advance a flexible treatment of different optimization objectives, and emphasize the effects of different factors on parallel performance effectively through the adjustable weighting parameters.

**Key words:** graph partitioning; parallel computing; load balancing; communication overhead

### 1 引言

目前, 大量文献提出了很多基于图划分的组合模型, 这些模型可以应用在并行计算中处理复杂和不规则的问题<sup>[1-3]</sup>、VLSI设计<sup>[4,5]</sup>、地理信息系统<sup>[6]</sup>、数据聚类<sup>[7,8]</sup>和图像分割<sup>[9,10]</sup>等. 在这些应用中, 把原始问题公式化为图划分问题, 努力优化某个目标函数 (例如: 最小化并行计算中总的通信量, 优化电路配置, 在处理GIS查询中最小化磁盘页的访问数), 而保持一定的限制 (例如: 并行系统中平衡计算负载, 使用磁盘页的能力作为数据分配的上界). 在并行计算中, 目标就是在处理器之间以一种减少通信代价和取得最大性能的方式来分配数据 (任务).

在最近的一些工作中, Catalyürek, Aykanat<sup>[12,13]</sup>和 Hendrickson<sup>[14]</sup>分别给出了图划分模型应用于并行求解稀疏矩阵与向量乘积时存在的一些缺点, 后来 Hendrickson, Kolda<sup>[11]</sup>给出了图划分模型应用于并行计算时同样存在下面一些缺陷:

#### 1.1 边割度量的缺点

(1) 边割不能精确的代表通信量. 如图1所示, 椭圆对应不同的处理器, 并行计算的任务用图中顶点表示, 图中的边表示任务间相互依赖关系, 在这个例子中, 任务 (顶点) 被划分到两个处理器上, 每条边的权值都为2 (假设每个方向产生一个单位的数据通信量), 因此, 所有割边权

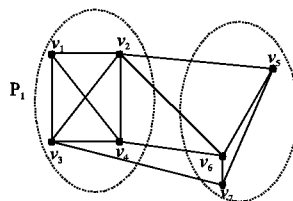


图1 边割对应通信量

Fig. 1 Edge cuts versus communication volume

值的和为8. 在并行计算过程中, 处理器  $P_1$  上的点  $v_2$  和  $v_6$  需要进行通信, 但处理器  $P_1$  与  $P_2$  只需要进行一次通信, 就能将顶点  $v_2$  的数据传送到  $P_2$ . 类似的还有点  $v_6$ , 这样, 实际的通信量是6. 一般来说, 边割不能识别两条或更多的边代表同一个信息

流, 因此过多地计算了实际的通信量.

(2) 在并行计算机上发送一个消息的时间跟延迟 (开始时间) 和消息大小有关. 图划分方法近似最小化总的通信量,

收稿日期: 2009-12-23 收修稿日期: 2010-03-30 基金项目: 国家自然科学基金项目 (60873256) 资助. 作者简介: 马永刚, 男, 1978 年生, 博士研究生, 研究方向为并行计算, 网络优化设计; 谭国真, 男, 1960 年生, 教授, 博士生导师, 研究方向为集群计算与网格计算, 智能交通系统; 杨际祥, 男, 1975 年生, 博士研究生, 研究方向为并行计算; 潘 东, 男, 1985 年生, 硕士研究生, 研究方向为并行计算.

而不是总的消息数. 依赖于机器的结构和问题的规模, 消息延迟可能比消息量更重要<sup>[11]</sup>.

(3) 没有考虑最慢处理器的情况, 并行应用的性能一般受到最慢处理器的限制. 例如, 即使计算达到了负载平衡, 但某个处理器需要跟很多邻居处理器进行通信, 其它的处理器就要等待这个“重通信量”的处理器完成所有的通信任务. 因此, 要考虑任何单个处理器所处理的最大消息数和消息量.

## 1.2 缺少表现性

(1) 无向图只能表示对称和双向依赖关系的应用, 对于非对称依赖关系和矩形矩阵的系统, 图划分模型不适用.

(2) 一般的计算框架是不断地重复进行相同的计算, 计算通常由几个不同的阶段组成, 多个阶段的计算一般不能经由无向图描述.

尽管图划分模型存在上面一些缺点, 但是无数的例子说明图划分模型对于并行求解微分方程, 稀疏矩阵迭代求解和基于网格的问题是非常有效的<sup>[11]</sup>. 因为, (1) 这些应用是典型的稀疏和规则的问题, 每个节点通常具有较少数目的邻居, 通信量比较均匀地分布到每个处理器, 所以图划分模型应用的较好. (2) 计算网格一般具有高度的几何位置性, 能够保证存在好的划分. 由于图划分模型成功地应用于许多领域, 并且存在很好的划分工具, 其中有高质量的串行 (Scotch<sup>[15]</sup>, Chaco<sup>[16]</sup>, Metis<sup>[17]</sup>, Jostle<sup>[18]</sup>, Party<sup>[19]</sup>) 和并行图划分工具 (PJostle<sup>[18]</sup>, ParMetis<sup>[20]</sup>), 所以图划分问题不断得到人们深入地研究.

许多并行计算问题归结为解决图划分问题<sup>[1-3]</sup>, 在并行计算的任务图中, 点代表计算任务, 边代表数据依赖关系, 点和边可以分配一定的权值, 分别表示计算的任务量和需要进行交换的数据量. 通过对任务图进行  $k$  路划分, 可以把任务分配到  $k$  个处理器上, 由于划分分配到每个处理器的任务的总权值近似相等, 也就是  $k$  个处理器上要完成的工作负载是平衡的. 划分的目标是使最小化边割的权值和, 也就是使通信量最小化.

为了使图划分模型更加有效地应用于无结构和不规则问题的并行计算, 需要修正前面提到的图划分模型存在的几个缺点, 因此, 我们给出一个改进的模型, 使用的代价函数能够克服基于边割度量代价的一些缺点, 可以有效地限制实际通信的额外开销, 并且能够以一种可以调整的方式对延迟, 整体通信额外开销和最大的局部通信额外开销进行优化调整, 使其更加适合实际的应用.

## 2 问题定义

### 2.1 图划分的定义

对于图  $G = (V, E)$ ,  $V$  为顶点的集合,  $E$  为边的集合,  $|V| = n$  为顶点的个数, 划分点集  $V$  为  $k$  个子集  $V_1, V_2, \dots, V_k$ , 对于任意的  $i, j = 1, \dots, k$ , 满足  $i \neq j, V_i \cap V_j = \emptyset$  和  $\cup_i V_i = V$ , 并且每个子集包含近似相等的顶点数, 而划分的边割数最小 (划分的边割定义为两个顶点属于不同划分区域的边), 这称作  $k$ -路图划分. 每个子集  $V_i$ , 叫做一个划分域或子域.  $V$  的一个  $k$ -路划分, 通常用一个长度为  $n$  的划分向量表示, 对每个点

$v \in V$ ,  $k[v]$  是一个介于 1 到  $k$  之间的整数, 表示点  $v$  属于的划分域.

如果每个点和边都具有权值, 那么图  $G$  是一个加权图. 加权图划分是把顶点分成  $k$  个不交的子集, 每个子集包含的点权和尽量相等, 而割边的权值和最小. 顶点集  $V$  的权值为  $V$  中所有点的权值和, 用  $w(V)$  表示. 划分域  $V_i$  的权值为  $V_i$  中所有点的权值和, 用  $w(V_i)$  表示, 所以

$$w(V) = \sum_{i=1}^k w(V_i)$$

图划分应用于并行计算时, 要完成的任务用一个图  $G = (V, E)$  表示, 点集  $V$  表示分配到的处理器上的对象 (例如: 元素, 节点等), 边集  $E$  描述了对对象间的依赖关系, 如果点  $i$  和点  $j$  被分配到不同的处理器, 并且它们之间共享信息, 那么它们就需要进行通信, 所以点  $i$  与  $j$  之间存在一条割边  $e_{ij}$ . 通过对任务图进行划分, 每个处理器得到近似相等的任务量, 而相互处理器间的通信量最小. 如果点和边都具有权值, 它们分别表示任务的计算量和通信量, 那么对加权图进行划分, 使得每个处理器具有近似相等的点权和 (负载均衡), 而子区域间所有边割的权值和最小 (也就是最小化整个通信量).

### 2.2 通信量和通信额外开销

并行计算就是把全体的计算任务分配到  $P$  个协同工作的处理器上, 在共享式处理器环境下, 各处理器通过互连网访问共享主存进行数据交换; 而分布式处理器环境下, 相互处理器间的通信通过互换消息实现. 当处理器的工作负载比较均衡并且相互处理器间通信的额外开销比较小的时候会取得好的并行性能. 这里主要考虑同构的分布式处理器环境下的并行计算.

#### 2.2.1 通信量的度量

首先考虑并行计算中必要的通信量, 划分域间的通信是以互换消息的形式实现的, 消息的形式一般为数值数组, 实际通信量是每个点所在处理器与邻居处理器间的通信量<sup>[11,14]</sup>.

用  $V_b \subset V$  表示边界点的子集, 也就是说对每个点  $v \in V_b$ ,  $v$  连接其它划分域中至少一个点. 对每个点  $v \in V_b$ , 用  $Nadj[v]$  表示点  $v$  的邻居划分域数, 那么点  $v$  产生的通信量可以表示为  $\sum_{v \in V_b} Nadj[v]$ , 称这个量为划分的边界割. 边界割对实际的应用是一个更加准确的度量<sup>[14]</sup>.

对于加权图来说, 如果点  $v$  发送的数据量为  $w_v$ , 那么点  $v$  产生的通信量为  $w_v Nadj[v]$ . 总的通信量可以表示为

$$totalv = \sum_{v \in V_b} w_v Nadj[v].$$

定义 1. 为了准确衡量划分域  $S_i$  的通信量, 定义划分域  $S_i$  的局部通信量为

$$C_{S_i} = \sum_{v \in S_i \cap V_b} Nadj[v], \quad (1)$$

在加权图中则为  $C_{S_i} = \sum_{v \in S_i \cap V_b} w_v Nadj[v]$ .

有了局部通信量的定义, 相应地, 可以给出全局通信量的定义.

定义 2. 全局通信量的定义为

$$C_{glob} = \sum_{i=1}^k C_{S_i}. \quad (2)$$

通信量是并行计算中考虑的一个重要的代价度量, 要想

获得高效率的并行计算,通信是至关重要的,提高性能的关键是减少通信量和通信次数。

### 2.2.2 量化通信额外开销

通过定义1和定义2能够精确的度量局部和全局的通信量,但是它们不能代表实际通信中的额外开销。因为通信的额外开销也要依赖于邻居处理器间通信的消息数和通信延迟。下面我们给出通信代价精确的表达形式。

假设互换  $M$  个数值的消息,花费下面的时间量,

$$\alpha + M\beta \quad (3)$$

$\alpha$  为平均延迟(平均开始时间),  $\beta$  为网络传输单位数值量的消息所用的时间,  $\alpha$  和  $\beta$  为两个硬件参数。使用式子(3),我们可以准确估计通信的额外开销。

定义3. 子区域  $S_i$  的局部通信额外开销为

$$O_{S_i} = D_{S_i}\alpha + C_{S_i}\beta \quad (4)$$

其中  $D_{S_i}$  为接收子区域  $S_i$  发送消息的划分域数,而  $C_{S_i}$  由定义1给出。

有了局部通信额外开销的定义,我们相应地可以给出全局通信额外开销的定义。

定义4. 全局通信的额外开销定义为

$$O_{glob} = D_{glob}\alpha + C_{glob}\beta \quad (5)$$

其中  $D_{glob} = \sum_{i=1}^k D_{S_i}$ ,  $C_{glob}$  由定义2给出。

从上面的两个定义我们可以看到,并行计算要取得好的性能要求  $O_{glob}$  尽可能小,而且  $O_{S_i}$  的最大值直接影响并行计算的执行时间,因为最慢的处理器决定了并行计算的时间。

并行计算中通信的额外开销不仅依赖于式子(5)中给定的  $O_{glob}$ ,而且依赖于(4)中给定的  $O_{S_i}$ ,它们由硬件参数  $\alpha$  和  $\beta$ ,以及  $D_{glob}$ ,  $C_{glob}$ ,  $D_{S_i,max}$  和  $C_{S_i,max}$  所决定,所以  $D_{glob}$ ,  $C_{glob}$ ,  $D_{S_i,max}$  和  $C_{S_i,max}$  的值应该尽可能的小,并且这些目标应该整合到一个统一的代价函数,才能够从整体上衡量各个量对并行性能的影响。

## 3 改进的图划分模型

为了解决传统图划分模型应用于并行计算时存在的一些问题,我们采用文献[21]中划分无结构有限元网格的思想并对其进行推广来解决图划分问题。下面我们给出改进的图划分模型,该模型能够把上面提到的各个目标统一到一个目标函数。

### 3.1 考虑延迟

对于划分更加合理的代价度量应该包括延迟的影响,也就是  $O_{glob}$  应该是最重要的优化目标,而不是  $C_{glob}$ 。由  $O_{glob}$  的定义我们知道,  $D_{glob}$  和  $C_{glob}$  应该尽可能的小。

式子(5)中具有两个硬件参数  $\alpha$  和  $\beta$ ,这样不方便  $O_{glob}$  直接作为一个衡量代价的函数。下面引进一个没有维数的参数  $\gamma$ ,它定义为

$$\gamma = \frac{\alpha}{\beta} \quad (6)$$

$\gamma$  是  $\alpha$  和  $\beta$  的比率,它可以作为一个相对的延迟代价,使用参数  $\gamma$ ,可以简化通信代价的表达形式。

定义5. 通过参数  $\gamma$  的定义,我们可以给出全局通信代价的定义

$$P_{glob} = \gamma D_{glob} + C_{glob} \quad (7)$$

从这个定义我们可以看到,  $P_{glob}$  和  $C_{glob}$  相差的是  $\gamma D_{glob}$ 。  $\gamma$  是一个可以调整的参数,我们可以通过增加  $\gamma$  来强调延迟对并行性能的影响,或者选择  $\gamma = 0$  来忽略延迟的影响。而且,我们很容易看到下面这个关系

$$P_{glob} = \gamma D_{glob} + C_{glob} = \frac{1}{\beta} (D_{glob}\alpha + C_{glob}\beta) = \frac{1}{\beta} O_{glob} \quad (8)$$

所以,  $P_{glob}$  能够反映  $O_{glob}$  的大小,而  $O_{glob}$  代表了全局通信的额外开销。

### 3.2 考虑通信额外开销的分布

通信额外开销在划分域间平衡的分布也是我们期望的一个目标。如果某个划分域占据了  $P_{glob}$  的大部分,那么其它的子域就要等待这个“重通信量”的子区域完成它的所有通信任务。因此,对于图划分一个合理的代价函数在最小化  $P_{glob}$  的同时,也应该最小化下面这个式子

$$P_{S_i,max} = \max_{1 \leq i \leq k} (\gamma D_{S_i} + C_{S_i}) \quad (9)$$

的值。式子(9)可以作为单个划分域最大通信代价的度量。

### 3.3 代价函数的统一公式

为了使图划分更加适合并行计算,上面的因素应该综合考虑并整合到一个代价函数,为此,我们给出一个统一的多目标代价公式:

$$C(\gamma, \lambda_{glob}, \lambda_{subd}) = \lambda_{glob} P_{glob} + \lambda_{subd} P_{S_i,max} \quad (10)$$

(其中  $\lambda_{glob} \geq 0$ ,  $\lambda_{subd} \geq 0$  且  $\lambda_{glob} + \lambda_{subd} = 1$ )。

我们可以看到,上面的代价函数  $C(\gamma, \lambda_{glob}, \lambda_{subd})$  既与反映通信延迟的参数  $\gamma$  有关,也与两个加权参数  $\lambda_{glob}$  和  $\lambda_{subd}$  有关。我们可以通过调整  $\lambda_{glob}$  和  $\lambda_{subd}$ ,很容易强调全局和最大的局部通信代价的影响。由于可以自由调整参数  $\gamma$ ,  $\lambda_{glob}$  和  $\lambda_{subd}$ ,公式(10)构成了一个多目标代价函数的统一模型。使用下面特殊的参数集合  $\gamma = 0, \lambda_{glob} = 1, \lambda_{subd} = 0$ ,就得到了标准图划分模型。

上面的代价函数,不仅考虑了通信延迟和通信量的精确度量,也考虑了单个划分域最大通信额外开销的分布对并行性能的影响,克服了标准图划分模型边割度量的一些缺点,而且可以通过调整加权参数强调各个量对并行性能的影响,所以,提出的改进模型更加适合并行计算的应用。

## 4 实验模拟与测试结果

为了验证我们提出的改进模型的合理性,我们通过下面的算法对其进行验证。算法的具体步骤如下:

**Step 1.** 通过常用的图划分软件 Metis 对测试集中随机选取的数据进行划分,分成  $k$  个部分( $k$  个处理器);

**Step 2.** 计算每个划分域的通信量和每个划分域的邻居划分域数;

**Step 3.** 计算总的通信量和总的邻居划分域数;

**Step 4.** 给定参数  $\gamma$ ;

**Step 5.** 计算所有子区域中最大的通信额外开销;

**Step 6.** 给定参数  $\lambda_{glob}, \lambda_{subd}$  的值;

**Step 7.** 得到不同参数时的代价函数,查看每个因素对并行性能的影响。

我们从图划分标准测试数据集<sup>[22]</sup>中随机选取了 add32, 3elt 和 finan512 三组数据进行实验,数据特征如表 1 所示。

表 1 数据特征

Table 1 Data character

数据名	点数	边数	最小 度数	最大 度数	平均 度数	描述
add32	4960	9462	1	31	3.82	32-bit adder
3elt	2851	15093	3	9	5.81	2D nodal graph
finan512	74752	261120	2	54	6.99	Portfolio optim

这三组数据所形成的图分别表示用于并行计算的任务图。每个任务图分别被划分成 4, 8, 16, 32 和 64 个部分,使用上面的算法,变换各个加权参数,获得相应的结果。为了比较各个参数的变化对代价函数的影响,我们选取的延迟参数与参数  $\lambda_{glob}$  和  $\lambda_{subd}$  为一个数量级,即  $\gamma$  为小于 1 的数,得到改进图划分模型中各个量的仿真实验结果如下:

表 2 数据集 add32 各个参数的值

Table 2 Data set add32 every parameter

$K$	$\gamma$	$\lambda_{glob}$	$\lambda_{subd}$	$D_{glob}$	$C_{glob}$	$D_{S_i,max}$	$C_{S_i,max}$	$C(\gamma, \lambda_{glob}, \lambda_{subd})$
4	0.3	0.3	0.7	69	138	28	58	94.09
4	0.5	0.3	0.7	69	138	28	58	102.15
4	0.3	0.5	0.5	69	138	28	58	112.55
4	0.3	0.1	0.9	69	138	28	58	75.63
8	0.4	0.6	0.4	106	198	19	40	162.64
8	0.7	0.6	0.4	106	198	19	40	183.52
8	0.4	0.9	0.1	106	198	19	40	220.96
8	0.4	0.3	0.7	106	198	19	40	104.32
16	0.3	0.4	0.6	242	448	33	54	246.58
16	0.5	0.4	0.6	242	448	33	54	269.90
16	0.3	0.6	0.4	242	448	33	54	337.92
16	0.3	0.2	0.8	242	448	33	54	155.24
32	0.2	0.5	0.5	328	536	32	49	328.00
32	0.4	0.5	0.5	328	536	32	49	363.50
32	0.2	0.7	0.3	328	536	32	49	437.44
32	0.2	0.3	0.7	328	536	32	49	218.56
64	0.5	0.3	0.7	769	1292	24	47	544.25
64	0.7	0.3	0.7	769	1292	24	47	593.75
64	0.5	0.5	0.5	769	1292	24	47	867.75
64	0.5	0.1	0.9	769	1292	24	47	220.75

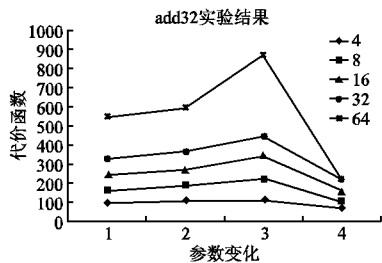


图 2 数据集 add32 的实验结果

Fig. 2 Data set add32 experiment result

为了比较各个参数对并行性能的影响,我们给出数据集 add32 对于不同参数取值变化时,得到的代价函数变化情况,如图 2 所示。

从这个图我们可以看到,任务被分成 4, 8, 16, 32 和 64 个

部分时,对参数  $\gamma, \lambda_{glob}$  和  $\lambda_{subd}$  的值进行变化,参数值的改变量类似的情况下,任务被划分成不同部分时的代价函数变化趋势是相似的。其余两组数据的实验结果也得到了类似的结论。

表 3 数据集 3elt 各个参数的值

Table 3 Data set 3elt every parameter

$K$	$\gamma$	$\lambda_{glob}$	$\lambda_{subd}$	$D_{glob}$	$C_{glob}$	$D_{S_i,max}$	$C_{S_i,max}$	$C(\gamma, \lambda_{glob}, \lambda_{subd})$
4	0.3	0.8	0.2	278	900	117	385	870.74
4	0.5	0.8	0.2	278	900	117	385	919.90
4	0.3	1.0	0.0	278	900	117	385	984.40
4	0.3	0.6	0.4	278	900	117	385	758.08
8	0.1	0.2	0.8	537	1616	108	332	608.18
8	0.3	0.2	0.8	537	1616	108	332	646.94
8	0.1	0.4	0.6	537	1616	108	332	873.56
8	0.1	0.0	1.0	537	1616	108	332	342.80
16	0.5	0.4	0.6	861	2636	107	335	1459.70
16	0.8	0.4	0.6	861	2636	107	335	1582.28
16	0.5	0.7	0.3	861	2636	107	335	2263.10
16	0.5	0.1	0.9	861	2636	107	335	656.30
32	0.2	0.3	0.7	1373	4054	78	229	1469.38
32	0.4	0.3	0.7	1373	4054	78	229	1562.26
32	0.2	0.5	0.5	1373	4054	78	229	2286.30
32	0.2	0.1	0.9	1373	4054	78	229	601.58
64	0.3	0.7	0.3	2372	6654	70	209	5224.92
64	0.6	0.7	0.3	2372	6654	70	209	5729.34
64	0.3	1.0	0.0	2372	6654	70	209	7365.60
64	0.3	0.4	0.6	2372	6654	70	209	3084.24

数据集 3elt 在参数取不同值时得到的代价函数变化见图 3。

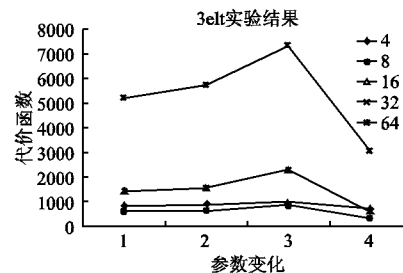


图 3 数据集 3elt 实验结果

Fig. 3 Data set 3elt experiment result

数据集 finan512 在参数取不同值时代价函数变化如图 4。

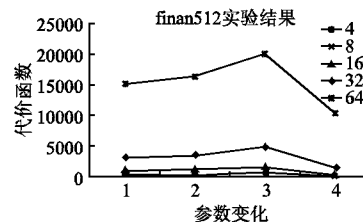


图 4 数据集 finan512 实验结果

Fig. 4 Data set finan512 experiment result

从上面的实验可以看到,对于划分域数(处理器数)较少的情况,通信延迟不会对代价函数产生较大的作用,全局通信额外开销是影响代价函数的主要因素,局部通信额外开销也是影响代价函数的一个要素。当划分域数较多的时候,通信延迟会对代价函数产生较大的影响,全局通信额外开销仍是影



响代价函数的主要因素,局部通信额外开销的变化同样会对代价函数产生较大的影响。

表4 数据集 finan512 各个参数的值

Table 4 Data set finan512 every parameter

$K$	$\gamma$	$\lambda_{glob}$	$\lambda_{subd}$	$D_{glob}$	$C_{glob}$	$D_{S_{ij},max}$	$C_{S_{ij},max}$	$C(\gamma, \lambda_{glob}, \lambda_{subd})$
4	0.6	0.3	0.7	296	648	78	162	393.84
4	0.9	0.3	0.7	296	648	78	162	436.86
4	0.6	0.6	0.4	296	648	78	162	578.88
4	0.6	0.0	1.0	296	648	78	162	208.80
8	0.4	0.2	0.8	592	1296	83	162	462.72
8	0.6	0.2	0.8	592	1296	83	162	499.68
8	0.4	0.4	0.6	592	1296	83	162	730.24
8	0.4	0.0	1.0	592	1296	83	162	195.20
16	0.2	0.3	0.7	1184	2592	86	162	974.08
16	0.5	0.3	0.7	1184	2592	86	162	1098.70
16	0.2	0.6	0.4	1184	2592	86	162	1768.96
16	0.2	0.0	1.0	1184	2592	86	162	179.20
32	0.3	0.5	0.5	2368	5184	81	162	3040.35
32	0.6	0.5	0.5	2368	5184	81	162	3407.70
32	0.3	0.8	0.2	2368	5184	81	162	4752.78
32	0.3	0.2	0.8	2368	5184	81	162	1327.92
64	0.1	0.6	0.4	10036	23828	232	550	15126.88
64	0.3	0.6	0.4	10036	23828	232	550	16347.60
64	0.1	0.8	0.2	10036	23828	232	550	19979.24
64	0.1	0.4	0.6	10036	23828	232	550	10274.52

## 5 结论和进一步研究

本文提出了一种改进的图划分模型,它的主要贡献在于:

1)解决了标准图划分模型存在的一些问题。在标准图划分模型中,边割度量不能精确代表通信量,没有考虑通信延迟的影响,也没有考虑最大通信额外开销的处理器对并行性能的影响。

2)本文采用统一的代价函数,把通信延迟和整体的通信额外开销,以及单个划分域最大通信额外开销统一为一个多目标代价函数,并且可以通过调整参数强调各个量对并行性能的影响。

3)可以通过调整目标函数,采用新的划分方法,使任务划分在保证负载均衡的基础上减少总体的通信代价。

今后的工作将包括:①进一步完善图划分的模型,克服标准模型中存在的缺陷,使其适合更加广泛的应用。②对改进的模型提出改进的算法。

## References:

- [1] Karypis G, Kumar V. Multilevel k-way partitioning scheme for irregular graphs[J]. Journal of Parallel and Distributed Computing, 1998, 48(1):96-129.
- [2] Horst D Simon. Partitioning of unstructured problems for parallel processing[J]. Computing Systems in Engineering, 1991, 2(2-3): 135-148.
- [3] Kumar V, Grama A, Gupta A, et al. Introduction to parallel computing: design and analysis of algorithms [M]. Redwood City, CA: Benjamin/Cummings Publishing Company, 1994.
- [4] Alpert Charles J, Kahng Andrew B. Recent directions in netlist partitioning: a survey[J]. Integration, the VLSI Journal, 1995, 19(1-2):1-81.
- [5] Lengauer T. Combinatorial algorithms for integrated circuit layout [M]. Chichester: Wiley-Teubner, 1990.
- [6] Guo J, Trinidad G, Smith N. MOZART: a multi-objective zoning and AggRegation tool [C]. Proceedings of 1st Philippine Computing Science Congress, 2000, 197-201.
- [7] Ding C, He X, Zha H, et al. A min-max cut algorithm for graph partitioning and data clustering[C]. Proceedings of the 2001 IEEE International Conference on Data Mining, 2001, 107-114.
- [8] Guénoche A. Comparing recent methods in graph partitioning[J]. Electronic Notes in Discrete Mathematics, 2005, 22:83-89.
- [9] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2000, 22:888-905.
- [10] Grady L, Schwartz E L. Isoperimetric graph partitioning for image segmentation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2006, 28:469-475.
- [11] Hendrickson B, Kolda T G. Graph partitioning models for parallel computing[J]. Parallel Computing, 2000, 26:1519-1534.
- [12] Catalyürek Ü V, Aykanat C. Decomposing irregularly sparse matrices for parallel matrix-vector multiplication [C]. Proceedings of 3rd International Workshop on Parallel Algorithms for Irregularly Structured Problems, 1996, 75-86.
- [13] Catalyürek Ü V, Aykanat C. Hypergraph-partitioning based decomposition for parallel sparse matrix vector multiplication [J]. IEEE Trans on Parallel Distrib. Syst, 1999, 10:673-693.
- [14] Hendrickson B. Graph partitioning and parallel solvers: has the emperor no clothes? [C]. LNCS 1457: Proceedings of the 5th International Symposium on Solving Irregularly Structured Problems in Parallel, 1998, 218-225.
- [15] Pelligrini F. SCOTCH 3.4 user's guide [R]. RR-1264-01, LaBRI, November 2001.
- [16] Hendrickson B, Leland R. The Chaco user's guide, version 2.0 [R]. SAND94-2692, Sandia National Laboratories, Albuquerque, NM, October 1994.
- [17] Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs [R]. CORR 95-035, University of Minnesota, Dept. Computer Science, Minneapolis, MN, June 1995.
- [18] Walshaw C. The parallel JOSTLE library user's guide, version 3.0 [R]. University of Greenwich, London, UK, 2002.
- [19] Preis R, Diekmann R. The PARTY partitioning library, user's guide, version 1.1 [R]. tr-rsf-96-024, Dept. of Computer Science, University of Paderborn, Paderborn, Germany, September 1996.
- [20] Karypis G, Kumar V. ParMETIS: parallel graph partitioning and sparse matrix ordering library [R]. Tech. Rep. 97-060, Department of Computer Science, University of Minnesota [EB/OL]. <http://www.cs.umn.edu/~metis>, 1997.
- [21] Cai X, Bouhmala N. A unified framework of multiobjective cost functions for partitioning unstructured finite element meshes [J]. Applied Mathematical Modelling, 2007, 31: 1711-1728.
- [22] Walshaw C. The graph partitioning archive [EB/OL]. <http://staffweb.cms.gre.ac.uk/~c.walshaw/partition/>, 2009.