

选择粗化函数优化并行稀疏矩阵向量乘法*

叶纬材

(中山大学数学与计算科学学院//广东省计算科学重点实验室, 广东 广州 510275)

摘要: 讨论了如何分划稀疏矩阵的非零元素以减少并行矩阵向量乘法的通信代价。通过以粗化函数为工具, 统一现有的数据分划方法; 提出一种基于行列分划为初解的粗化函数选取方法, 在理论上的证明其运行效率与分划质量不逊于一维数据分划方法; 实验数据表明, 该方法产生分划质量超过一维数据分划方法的结果, 接近甚至超过二维细粒度方上法的结果。

关键词: 选择粗化函数; 数据分划; 并行; 稀疏矩阵向量乘法

中图分类号: O241.83 **文献标志码:** A **文章编号:** 0529-6579 (2012) 05-0050-04

Optimizing Parallel Sparse Matrix-Vector Multiplication by Selected Contraction Functions

YE Weicai

(School of Mathematics and Computational Science//Guangdong Province Key Laboratory of Computational Science, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: A new method is presented for distributing data in sparse matrix-vector multiplication by selected contraction functions. And the contraction functions are selected. The quality and the complexity of this method are theoretically ensured not to worse than those of traditional one-dimensional partitioning methods. Experimental results show that this method often produces better results than one-dimensional methods and is competitive with the best two-dimensional methods.

Key words: selected contraction function; data partitioning; parallel; sparse matrix-vector multiplication

稀疏矩阵向量乘法是大量迭代计算方法的核心。这些方法常用于求解包括线性方程组、奇异值分解、以及数据搜索引擎的网页排序等问题。在迭代过程中, 稀疏矩阵保持不变, 而矩阵向量乘法被反复计算。提高并行稀疏矩阵向量乘法的效率就要解决稀疏矩阵的非零元素和输入输出向量元素的分布。并行化过程中出现的矩阵元素和输入输出向量的分划问题被称为并行稀疏矩阵向量乘法所对应的数据分划问题。这是一个重要的组合数学问题, 以下简称数据分划问题。

对数据分划问题, 我们只关心矩阵的稀疏模式 (sparsity pattern), 即矩阵的元素 a_{ij} 值非 0 即 1。

记 $Z_n := \{0, 1, \dots, n-1\}$ 为指标集, n 是个正整数。令矩阵等同于一个指标对集合, 记为 $A := \{(i, j) : i \in Z_m, j \in Z_n\}$ 。记非零元素个数为 $\text{nnz}(A)$ 。一个矩阵 A 的 p 部分划 (p -way partition) 是一个以 p 个 A 的非空子集合为元素的集合, 记为 $\{A_0, A_1, \dots, A_{p-1}\}$ 。这 p 个子集合是互不相交的, 而且满足 $\cup_{i \in Z_p} A_i = A$ 。每个 A_i 称为 A 的一部分 (part)。记所有矩阵 A 的非零元素对应的指标对组成的集合为 A^1 。由于数据分划问题只与矩阵的非零元素有关, 并与矩阵的分划一一对应, 下面就用集合 A^1 的数据分划定义矩阵 A 的数据分划。

下面介绍并行稀疏矩阵向量乘法的计算过程。

* 收稿日期: 2011-09-07

基金项目: 广东省计算科学重点实验室资助项目; 计算科学科研团队资助项目

作者简介: 叶纬材 (1980年生), 男, 博士后; E-mail: cai_rey@yahoo.com.cn

假设共有 p 个计算单元，输入矩阵 A 为一个 $m \times n$ 的矩阵。每个计算单元拥有矩阵 A 的非零元素以及输入输出向量元素的一部分。拥有矩阵非零元素 a_{ij} 的计算单元求出乘积 $a_{ij}x_j$ 。若 a_{ij} 、 x_j 和 y_i 属于同一个计算单元，则计算过程为本地的，否则就需要通信。一个自然的并行稀疏矩阵向量乘法由以下四个步骤组成：

1) 每个计算单元发送其拥有的向量元素 x_j 到其他拥有第 j 列非零元素的计算单元；

2) $y_{ik} \leftarrow \sum_{a_{ij} \in A_k} a_{ij}x_j$ ，其中 y_{ik} 为第 i 行在计算单元 k 上的部分和， k 是计算单元的序号，从 0 到 $p-1$ ；

3) 第 k 号计算单元发送非零的 y_{ik} 到拥有 y_i 的计算单元上；

$$4) y_i \leftarrow \sum_{k=0, y_{ik} \neq 0}^{p-1} y_{ik}.$$

任意计算架构下和任意数据分划条件下的并行稀疏矩阵向量乘法都可以由以上的算法演变而成。

计算过程中，所有计算单元需要的非本地向量元素个数的总和称为**通信量**。通信量与矩阵和输入输出向量的分划相关。我们假设向量的分划采用非对称方式，并由矩阵分划导出。这样，数据分划问题就只和矩阵的数据分划有关。

本文讨论的数据分划问题定义如下：给定一个稀疏矩阵 A 和一个正整数 p ，寻找 A^1 的一个 p 部分划，不但要满足负载平衡条件

$$\max_{i \in Z_p} \{ \text{nnz}(A_i) \} < \frac{\text{nnz}(A_i)(1 + \varepsilon)}{p} \quad (1)$$

而且通信量要低。常数 $\varepsilon \in [0, 1)$ 称作**负载不平衡因子**，表示最大可接受的计算负载不平衡程度。

通信量在分布式集群的节点之间表现为网络带宽的消耗量，而在节点内部表现为内存带宽的消耗量。在多核计算架构下，通过数据分划，不但减少内存带宽的消耗量，而且降低内存争用，从而进一步提高并行矩阵向量乘法的效率。

文章的结构如下：第一节以粗化函数的观点统一现有的数据分划方法。第二节给出一种新的自适应粗化函数构造方法，并给出该方法对应解空间的理论性质。第三节是分划实验。最后我们做一总结。

1 粗化函数的选择与一维、二维矩阵分划

在本节里，我们用粗化函数的观点统一回顾现有的数据分划方法。首先通过分析矩阵分划解的结构，导出求解数据分划问题的过程就是寻找粗化函

数的过程；然后提出一类显式粗化函数选择的方法、作用、以及与现有一维、二维矩阵分划方法的关系。

首先介绍矩阵数据分划问题解的结构。每一个矩阵 A 的 p 部分划都对应一个函数 $f: A^1 \mapsto p$ ，记为 $f(A, i, j)$ ， $(i, j) \in A^1$ 。函数 f 被称为**矩阵分划映射函数**。矩阵 A 的每一部分都可以表示成

$$A_k := \{ (i, j) : f(A, i, j) = k, (i, j) \in A^1 \}, k \in Z_p,$$

矩阵分划映射函数 f 都有以下的结构

$$f(A, i, j) = c \circ \iota(i, j), (i, j) \in A^1 \quad (2)$$

函数 $\iota: A^1 \mapsto Z_{\text{nnz}(A)}$ 是一个双射，将每个非零元素进行排列并编号。号码从 0 到 $\text{nnz}(A) - 1$ 。函数 $c: Z_{\text{nnz}(A)} \mapsto Z_p$ 是一个满射，指出每个非零元素所属部分的号码。函数 ι 被称为**序列化函数** (serialization function)，而函数 c 被称为**粗化函数** (contraction function)。由此可见，求解数据分划问题的过程就是寻找粗化函数的过程。

这种将每个非零元素独立分划的方法称为**二维细粒度矩阵分划**。该问题可以建模为**细粒度超图分划问题** (fine-grained hypergraph partitioning problem)，由文 [1] 提出。

文 [2-7] 等都作了不同方面的改进。由于一般超图分划问题是 NP-hard 问题^[8]，所以求解以上的问题的算法多数是启发式的。文 [1] 提出的近似算法是现存文献中平均性能最优的，其多层迭代方法计算复杂度^[9] 约为 $(\text{nnz}(A) \log(\text{nnz}(A)))$ 。为了加速计算过程，最直接的方式就是对粗化函数的选择加上约束，以降低求解问题的复杂度。

下面讨论一类通过特殊粗化函数导出约束分划问题解的方法。令 c_0 为一个给定的、将 $Z_{\text{nnz}(A)}$ 映射到 Z_{k_0} 上的满射，其中 $k_0 > p$ 。如果对应于矩阵 A 的 p 部数据分划问题的解 f 都有下面的形式

$$f(A, i, j) = c \circ c_0 \circ \iota(i, j), (i, j) \in A^1 \quad (3)$$

约束就通过 c_0 显式地添加到解空间里面。这里的 $c: Z_{\text{nnz}(A)} \mapsto Z_{k_0}$ 是任何一个粗化函数。附加的约束表现为

$$c_0 \circ \iota(i_1, j_1) = c_0 \circ \iota(i_2, j_2) \Rightarrow f(A, i_1, j_1) =$$

$$f(A, i_2, j_2), \forall (i_1, j_1), (i_2, j_2) \in A^1 \quad (4)$$

记函数空间为 $F(A, c_0 \circ \iota, p)$ ，其包含所有关于矩阵 A 的 p 部分划函数 f ，而且这个空间是由函数 $c_0 \circ \iota$ 所引进的。该函数空间具体定义成以下形式

$$F(A, c_0 \circ \iota, p) := \{ f : f(A, i, j) = c \circ c_0 \circ \iota(i, j), (i, j) \in A^1 \} \quad (5)$$

引理 1 令 f 为一个关于矩阵 A 的 p 部分划问题的分划映射函数。 $f \in F(A, c_0 \circ \iota, p)$ 当且仅当 f

满足条件 (4)。

例如文 [10] 中讨论的矩阵的一维 (1-D) 行分划问题加入了粗化函数 c_{row} , 保证同一行的非零元素将分属同一部分。这个函数的形式为

$$c_{row} \circ \iota(i, j) = i, (i, j) \in A^1 \quad (6)$$

同理, 矩阵的 1-D 列分划问题加入了粗化函数 c_{col} 。这个函数的形式为

$$c_{col} \circ \iota(i, j) = j, (i, j) \in A^1 \quad (7)$$

我们称函数 c_{row} 和 c_{col} 为 1-D 粗化函数。由于引入了约束, 1-D 分划解空间就比二维 (2-D) 细粒度分划问题的要小的多; 但同时分划的质量有可能变差。尽管 2-D 细粒度分划方案要灵活的多, 但是对大型矩阵来说, 计算的规模就增加不少。

近来出现了新的粗化函数选择方法。2-D Mondriaan 方法基于分部 1-D 分划^[3], 往往得到比传统 1-D 方法优胜的结果。角型分划方法基于粗化函数 $c_{corner-row}$ 和 $c_{corner-col}$ ^[4]。两个函数的具体形式为

$$c_{corner-row} \circ \iota(i, j) = \min(i, j), (i, j) \in A^1 \text{ 和}$$

$$c_{corner-col} \circ \iota(i, j) = \max(i, j), (i, j) \in A^1$$

对箭头型矩阵等例子, 角型方法得到的结果是可以与传统的 2-D 方法的结果媲美; 但也有部分例子, 得到的结果比 1-D 方法的结果更差。以上的两种方法的计算速度都比 2-D 细粒度方法要快得多。

2 粗化函数的选择与分片一维分划

在本节里, 我们提出一种新的自适应粗化函数构造方法。目标是提出一种比 1-D 方法更灵活但计算复杂度相若的方法。为了简化讨论, 我们在本节中只关注 2 部分划问题。

令 $\{A_0^*, A_1^*\}$ 为矩阵 A 的一个具体的行分划结果。又令 C_{A^*} 是 Z_n 的一个非空子集合。假设 C_{A^*} 包含了所有引发通讯的列的序号。引入一个新的粗化函数 $c_{new}: Z_{n \times n(A)} \mapsto Z_{|C_{A^*}|+n}$ 。记为

$$c_{new} \circ \iota(i, j) = \begin{cases} j, & j \in Z_n \setminus C_{A^*}, \\ j, & j \in C_{A^*} \text{ and } (i, j) \in A_0^*, \\ \nu_{new}(j), & j \in C_{A^*} \text{ and } (i, j) \in A_1^* \end{cases} \quad (8)$$

其中 ν_{new} 是一个序列化函数, 将每个 $j \in C_{A^*}$ 进行编号, 从 n 到 $|C_{A^*}|+n-1$ 。因为 C_{A^*} 是 Z_n 的非空子集合, 所以 $|C_{A^*}| \leq n$ 。因此, c_{new} 导出的解空间的规模与 1-D 方法相近。

下面的引理和定理给出了新方法在理论上的优

势。

引理 2 所有的列分划解都属于 c_{new} 导出的解空间, 即

$$F(A, c_{col} \circ \iota, 2) \subset F(A, c_{new} \circ \iota, 2)$$

定理 1 如果选择 1-D 行分划的最优解 $\{A_0^*, A_1^*\}$, 函数空间 $F(A, c_{new} \circ \iota, 2)$ 里解的最小通信量不大于 1-D 行分划问题解的最小通信量。

因此, c_{new} 导出的分划映射解本质是一种分片一维分划, 并且有以下优势:

(i) 选取最优的行分划初解, 质量不比 1-D 方法差;

(ii) 选择性粗化方法的粗化函数是动态自适应选择的, 这与“角方法”不同, 灵活度提高;

(iii) 分划的粒度比“Mordiaan 方法”要小得多, 有利于找到更好的解;

(iv) 计算复杂度与 1-D 方法相若。

3 实验

在本节中, 我们将比较各种算法的分划质量。参与比较的算法包括本文提出的选择性粗化分划算法 (Selective)、1-D 超图行分划方法^[10]、2-D Mondriaan 方法^[3]和 2-D 细粒度超图分划方法^[1]。负载不平衡因子设定为 0.03。所有的实验都运行 20 次, 取最佳结果。

实验在 Intel Core T2050 处理器下运行, 内存容量为 1GB。实验操作系统: Ubuntu Linux 8.04。每一个程序都用单进程和单核心, 独占系统运行。

测试用的矩阵都是在网络上可以取得的, 选自不同的问题领域。具体的矩阵信息, 请参阅表 1。

表 1 测试用矩阵的性质

Table 1 Properties of matrices for test

名称	行数 / 列数	非零元素个数	应用领域
cage10	11, 397	150, 645	DNA 电泳
Dubcova2	65, 025	1, 030, 225	PDE 求解
garon2	13, 535	373, 235	计算流体力学
lhr34	35, 152	764, 014	化学工程
orani678	2, 529	90, 158	经济问题

由表 2 所列的数据表明, 2-D 细粒度超图分划方法的平均质量是最好的。这是由于其分划空间理论上各种方法中最大的。本文提出的选择性粗化分划方法也得到了很不错的效果。对矩阵 cage10、Dubcova2 和 garon2, 选择性粗化分划方法得到的结果胜过所有其他方法的结果, 这就验证了

选择性粗化分划方法其分划空间在理论上要比 1 - D 分划方法要好的结论。表 2 中 (*) 表示结果

表 2 分划结果的通信量
Table 2 The communication volume of partitioning results

算法	份数	cage10	Dubcova2	garon2	lhr34	orani678
2 - D 细粒度超图分划 4	4	4 061	1 375	1 380	571	1 056
	16	8 636	4 296	4 112	1 907	2 785
	64	16 469	10 350	9 974	6 814	5 595
	256	29 557	23 965	21 573	25 611	10 915
选择粗化	4	4 094	1 314	1 005	323	1 083
	16	8 276	3 873	2 829	2 585	3 132
	64	17 110	9 964	7 063	6 981	6 665
	256	31 073	22 753	16 463	26 797	11 171
2 - D Mondriaan	4	4 949	2 634	1 109	865	1 145
	16	11 240	6 982	3 536	2 513	3 398
	64	19 306	14 471	9 484	9 770	8 714
	256	31 750	29 872	26 088	29 133	19 626
行分划	4	5 419	1 845	1 158	323	1 159
	16	12 553	5 413	3 547	1 819	5 220
	64	23 321	12 937	8 626	7 356	19 369
	256	40 740	29 515	20 461	25 955	*

4 总 结

本章提出以粗化函数的观点统一现有的数据分划方法，并且提出一种新的粗化函数选择方法。通过理论分析与实验证明，本文提出的选择性粗化方法比 1 - D 分划方法好，接近甚至超过 2 - D 细粒度方法的结果。

参考文献：

[1] CATALYUREK U, AYKANAT C. A fine-grain hypergraph model for 2D decomposition of sparse matrices [C]//Proceedings of 15th International Parallel and Distributed Processing Symposium(IPDPS), San Francisco, CA, 2001.

[2] CATALYUREK U, AYKANAT C. A hypergraph-partitioning approach for coarse-grain decomposition [C]//ACM/IEEE SC2001, Denver, CO, 2001.

[3] VASTENHOUW B, BISSELING ROB H. A two-dimensional data distribution method for parallel sparse matrix-vector multiplication [J]. SIAM Review, 2005, 47(1): 67 - 95.

[4] WOLF M M, BOMAN E G, HENDRICKSON B. Optimizing parallel sparse matrix-vector multiplication by corner partitioning [C]//PARA08, Trondheim, Norway, 2008.

[5] CATALYUREK UMIT V, BOMAN ERIK G. Hypergraph-based dynamic load balancing for adaptive scientific computations [C]//Proceedings of IPDPS'07, Best Algorithms Paper Award, 2007.

[6] BOMAN ERIK G. A nested dissection approach for sparse matrix partitioning [C]// Proceedings in Applied Mathematics and Mechanics, 2008.

[7] HENDRICKSON B. Graph partitioning and parallel solvers; has the emperor no clothes? [J]. Lecture Notes in Computer Science, 1998, 1457: 218 - 225.

[8] LENGAUER T. Combinatorial algorithms for integrated circuit layout [M]. UK: Wiley, 1990.

[9] SAAB Y. Combinatorial optimization by dynamic contraction [J]. Journal of Heuristics, 1997, 3: 207 - 224.

[10] CATALYUREK U, AYKANAT C. Hypergraph-partitioning based decomposition for parallel sparse-matrix vector multiplication [J]. IEEE Transaction on Parallel and Distributed Systems, 1999, 10(7): 673 - 693.