

# 基于改进图划分的异构并行计算模型设计

袁再龙

(贵州师范学院 教育信息中心, 贵阳 550018)

**摘要:** 为了实现大规模计算机集群上的高效分布式并行计算, 设计了一种基于改进图划分和量子遗传算法的异构节点并行计算模型; 首先, 介绍了传统图划分模型并分析了其不足, 然后从图的有向性、通信开销计算和负载均衡度等方面对传统的图划分模型进行了改进, 从而得到一个改进的图划分模型; 最后, 以最小化通信开销和优化资源负载均衡为目标, 通过设计编码方案, 在改进的图划分模型上提出了采用量子遗传算法获取最优任务划分方案的最优解; 仿真实验表明: 文中方法能有效实现任务的并行计算, 与其它方法相比, 具有较小的通信开销和较好的负载均衡度, 具有很强的可行性。

**关键词:** 图划分; 任务; 并行计算; 负载均衡

## Design of Parallel Computing Based on Improved Graph Partitioning

Yuan Zailong

(Educational Information and Network Center, Guizhou Normal College, Guiyang 550018, China)

**Abstract:** In order to realize the effective distribute parallel computing in large computer group, a parallel computing model based on improved graph partitioning and quantum genetic algorithm was proposed. Firstly, the traditional graph partitioning model was analyzed and the defects were listed, then the graph partitioning model was improved by changing the direction, communication consumption and load balance and etc, then the improved graph partitioning was obtained. Finally, the coding scheme was designed by minimizing the communication consumption and optimizing resource load balance as the goal, the optimum solution was got by operating the quantum genetic algorithm. The simulation shows the method in this paper can realize task parallel computing, and compared with the other methods, it has less average locating error, and therefore, it has big feasibility.

**Key words:** graph partitioning; task; parallel computing; load balance

## 0 引言

随着微电子技术、晶体管技术和无线通信技术的发展, 软件和数据可以连接在并发执行的多台机器上获得海量的计算能力<sup>[1-2]</sup>。并行计算<sup>[3]</sup>是指将顺序执行的计算任务划分为若干可以同时执行的子任务, 将这些子任务分配到  $K$  个协同工作的处理器上, 并发执行这些子任务, 从而在完成整个计算任务的同时能实现最小的通信代价和最大计算性能<sup>[4]</sup>。

为了获得高效的并行计算性能, 如何在各处理器上分配任务成为了研究重点。现有的对并行计算进行研究的主要工作有: 文献 [5] 采用基于 Petri 网的 IOPT 网络对系统建模, 将计算任务分解成若干个子任务并行进行处理。文献 [6] 提出了一种基于  $K$ -Medoids 算法和 MapReduce 的并行算法。文献 [7] 设计了一种新的 MapReduce 分布式并行计算模型。文献 [8] 采用 MPI 并行编程模型对传统的聚类算法进行改进, 设计了一种基于蚁群算法和  $K$ -Means 算法的聚类算法, 通过引入 MPI 模型, 大大提高了聚类的效率。文献 [9] 提出了一种基于多层  $K$  路划分算法的并行任务分配策略。

上述工作具有重要意义, 但没有考虑到异构计算节点对任务分配的影响, 本文在上述工作的基础上设计了一种基于改进图划分的异构并行计算模型, 并通过实验证明了其有效性。

## 1 图划分模型

图划分<sup>[10]</sup>能实现将原始问题转换为图划分问题, 在满足一定约束的前提下优化某个目标函数来实现原问题的求解。目前已经在 VLSI 设计、图像分割和不规则问题中获得高度关注。

图划分模型可以定义为一个四元组  $G = (V, E, n, m)$ , 其中  $V$  为顶点集合,  $E$  为无向边集合,  $|V| = n$  为顶点个数,  $|E| = m$  为边的个数, 将  $V$  划分为  $k$  个互不相交的子集  $V = \{V_1, V_2, \dots, V_k\}$ , 对于任意的两个子集  $V_i$  和  $V_j$ , 其中  $i \neq j$ , 均包含近似相等的顶点, 将划分为的子集组称为  $V$  上的  $k$  路划分, 每个子集  $V_i$  均称为一个划分或子域。

(1)  $k$  路划分: 四元组  $G = (V, E, n, m)$  的  $k$  路划分可以通过一个长度为  $n$  的向量  $S$  表示,  $S$  中的每个元素  $S[i], 1 \leq i \leq n$  的值为  $1 \leq S[i] \leq k$ 。

(2) 外部邻接点: 对于任意节点  $v \in V_i$ , 寻找与其相连的所有的边  $e \in E$ , 这些边对应的另一个端点的集合称为节点  $v$  的邻居节点集  $Adj(v)$ , 假设  $v$  的邻居节点集  $Adj(v)$  中的任意节点  $u$  位于的区域  $V_j$ , 判断  $u$  是否与  $v$  位于同一个划分区域中, 即  $S(v)$  是否等于  $S(u)$  或  $i$  是否等于  $j$ , 如果满足, 称为  $u$  为  $v$  的内部邻接点, 否则  $u$  为  $v$  在区域  $V_j$  的外部邻接点。

(3) 基数: 对于任意节点  $v \in V_i$ , 其邻接点集  $Adj(v)$  中外部邻接点的个数  $Count(u \in Adj(v) \wedge S[u] \neq S[v])$  称为节点  $v$  的基数记作  $N(v)$ 。

收稿日期: 2013-11-19; 修回日期: 2014-01-13。

作者简介: 袁再龙(1978-), 男, 贵州思南人, 实验师, 主要从事计算机科学与技术和现代教育技术方向的研究。

## 2 改进图划分并行计算模型

### 2.1 图划分模型的缺陷

传统的图划分并行计算模型如图 1 所示, 其中  $V_1$  和  $V_2$  为两个不同的图划分, 同时也对应了两个不同的处理节点,  $V_1$  包含的顶点集为  $\{v_1, v_2, v_3, v_4\}$ ,  $V_2$  包含的顶点集为  $\{v_5, v_6, v_7\}$ , 其中每个顶点对应了并行计算中的任务, 边对应了任务之间的相互依赖关系, 如图 1 所示。

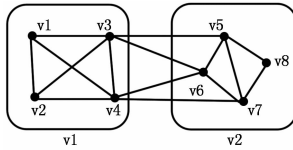


图 1 图划分模型示意

图划分模型用于并行计算时存在下述问题<sup>[11]</sup>:

- (1) 采用  $v$  的基数  $N(v)$  与边权值的乘积作为节点  $v$  的通信开销;
- (2) 图中的边均为有向边, 只能表示任务之间存在双向依赖的情况, 不能表示任务之间存在单向依赖的情况;
- (3) 为了实现处理节点负载均衡, 在每个处理节点上分配相同的任务量, 没有考虑节点的异构性, 即节点的计算、存储和传输带宽性能差异, 使得并行计算的性能受到性能最差处理节点的影响。

(4) 某区域中的一个节点与其它区域中几个节点通信时,

### 2.2 改进的图划分并行计算模型

定义 1: 改进的图划分并行计算模型可以表示为一个九元组  $G = (V, E, n, m, W, S, N, Adj, F)$ , 其中:

- (1)  $V$  为顶点集合, 每个顶点  $v \in V$  均对应了任务集中可以并发执行的子任务,  $V$  可以划分为  $k$  个不相交的子集  $V = \{V_1, V_2, \dots, V_k\}$ ;
- (2)  $E$  为有向边集合, 对于  $E$  中的任意一条边  $e \in E$ , 且  $e = (u, v)$ ,  $u$  为源节点,  $v$  为目标节点:  
当边  $e = (u, v)$  的两个顶点在同一个  $V_i$  中时, 表示  $v$  必须等待  $u$  执行完成才能开始工作, 即  $u$  和  $v$  将产生等待开销;  
当  $e = (u, v)$  的两个顶点在同一个  $V_i$  中时, 表示  $u$  和  $v$  将产生通信开销;
- (3)  $n$  为顶点个数即可并行执行的任务总数,  $m$  为边的条数, 表示有依赖关系的任务总对数;
- (4)  $Q(u)$  表示任务  $u$  的总计算量大小;
- (5)  $W(u, v)$  称为边  $e = (u, v)$  的权值, 表示边  $e = (u, v)$  上传送的消息量, 当顶点  $u$  和  $v$  位于同一区域时  $W(u, v) = 0$ ;
- (6)  $S$  为  $k$  路划分向量,  $N(v, V_i)$  表示节点  $v$  在区域  $V_i$  中的邻接点个数,  $Adj(v, V_i)$  表示节点  $v$  在区域  $V_i$  中的邻接点集;

(7)  $F(v)$  为任务  $v$  的通信开销, 假设区域  $V_i$  与区域  $V_j$  之间的单位通信开销为  $C(V_i, V_j)$ , 则  $F(v)$  可以通过下式计算:

$$F(v) = \sum_{v \in V_i} \sum_{\substack{u \in Adj(v, V_j) \\ v \in V_j, \wedge u \in V_i, \wedge V_i \neq V_j}} W(u, v) \cdot C(V_i, V_j) \quad (1)$$

### 2.3 并行计算目标函数

并行计算的目标即将  $n$  个任务分配到  $k$  个处理器上, 在实

现最小化通信开销的同时, 保证处理节点的负载均衡。由于任务只有在不同区域即不同处理节点上运行且为依赖型任务时才会产生通信开销, 因此, 只考虑依赖任务位于不同区域的情况, 由于式 (1) 对区域  $V_i$  节点的通信总开销进行计算, 因此, 在此基础上并行计算的目标就是使得所有节点的总开销最小, 且满足处理节点负载均衡, 目标函数定义为:

$$\begin{aligned} Goal = \min & \sum_{i=1}^k \sum_{v \in V_i} \sum_{u \in Adj(v, V_j)} W(u, v) \cdot C(V_i, V_j) \\ s. t. & \left| \sum_{v \in V_i} \frac{Q(v)}{R(V_i)} - \frac{1}{k} \sum_{j=1}^k \sum_{u \in V_j} \frac{Q(u)}{R(V_j)} \right| \leq \\ & \frac{1}{k} \sum_{j=1}^k \sum_{u \in V_j} \frac{Q(u)}{R(V_j)} \delta \quad (1 \leq i \leq k) \end{aligned} \quad (2)$$

其中:  $\delta$  为负载均衡参数,  $R(V_i)$  为处理节点  $V$  的处理能力, 约束部分对应了负载均衡约束。

## 3 基于改进量子遗传算法的并行计算

在获得任务和处理器对应的图划分九元组后, 采用改进的量子遗传算法实现任务的重分配方案搜索, 直到获取全局最优解作为最终的并行计算任务分配结果。

采用改进量子遗传算法实现并行计算的算法可以描述为:

初始化: 种群规模  $M$ , 当前迭代次数  $t$ , 参数  $\epsilon = 0.4$  迭代次数最大值  $T$ , 全局最优解  $S^* = [0, 0, \dots, 0]$ ;

步骤 1: 随机生成初始种群,  $p(t) = \{p'_1, p'_2, \dots, p'_n\}$  为含有  $M$  条染色体的种群, 其中的每个个体  $p'_i$  均可以编码为:

$$\begin{pmatrix} \left| \begin{array}{c|c|c|c} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1s} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1s} \\ \hline \alpha_{21} & \alpha_{22} & \dots & \alpha_{2s} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2s} \\ \hline \dots & \dots & \dots & \dots \\ \hline \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{ns} \\ \beta_{n1} & \beta_{n2} & \dots & \beta_{ns} \end{array} \right\rangle \end{pmatrix} \quad (3)$$

其中:  $k$  表示区域数,  $n$  为任务总数, 其中, 每个行向量中仅有一个  $(\alpha_{ij}, \beta_{ij})^T$  为 1 表示任务  $i$  划分处理节点  $j$  上, 即划分在  $V_j$  组。

步骤 2: 对每个染色体的每行的任意一个基因中量子比特  $(\alpha_{ij}, \beta_{ij})^T$  的初始化为  $(1/\sqrt{2}, 1/\sqrt{2})^T$ ;

步骤 3: 将染色体上量子比特门转换为任务分配向量  $S$ , 即随机产生一个  $0 \sim 1$  之间的随机数  $r$ , 取出染色体第  $j$  行中出现的第一个  $|\alpha_{jk}|^2 \leq r$  时的  $k$ , 令  $S[j] = k$ ;

步骤 4: 采用公式 (2) 求取种群  $p(t)$  中个体的适应度值, 记录最优解, 判断其是否比历史最优解  $S^*$  更优, 如果大于则采用当前最优解替换历史最优解; 否则, 保持当前最优解  $S^*$  不变;

步骤 5: 采用改进的量子旋转门和对所有量子比特进行更新, 以获取新解:

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (4)$$

其中:  $(\alpha, \beta)$  和  $(\alpha', \beta')^T$  分别为调整前和调整后的量子比特值。

步骤 6：当前迭代次数  $t = t + 1$ ，并判断当前迭代次数  $t$  达到最大值，如达到最大值，则输出最优解；否则转入步骤 3 进行继续迭代。

### 4 仿真实验

采用 Matlab 仿真工具进行实验，采用标准数据库<sup>[12]</sup>中的 6 个数据集进行实验，生成了对应的 6 幅图，图节点数分别为 50、80、120、150、180、200，节点权值为 10~300，节点间的单位通信量为 100~500 Bytes，边的权值为 5~35，负载均衡系数为 0.02~0.06，如表 1 所示。

表 1 图详细信息

图名称	节点数	点权值	边权值	负载均衡系数
图 1	50	10~300	5~35	0.02~0.06
图 2	80	10~300	5~35	0.02~0.06
图 3	120	10~300	5~35	0.02~0.06
图 4	150	10~300	5~35	0.02~0.06
图 5	180	10~300	5~35	0.02~0.06
图 6	200	10~300	5~35	0.02~0.06

采用文中改进的图划分方法对表 1 所示的图 1 到图 4 进行 3 路、4 路和 6 路划分即将所有节点划分到 3 个、4 个和 6 个区域中，而对图 5 和图 6 进行 4 路、5 路和 6 路划分，然后通过改进的量子遗传算法对通信开销和负载均衡度进行计算，与经典的 Kmetis 方法以及文献 [13] 方法进行比较，得到的结果如表 2 所示。

表 2 并行计算结果比较

图号	划分区 域数	经典的 Kmetis 方法		文献 [13] 方法		文中方法	
		通信 开销 (Mbyte)	负载均 衡度	通信 开销 (Mbyte)	负载均 衡度	通信 开销 (Mbyte)	负载均 衡度
1	3	2 563	1.02	2 156	1.03	2 132	1.03
1	4	3 103	1.04	2 942	1.03	2 313	1.02
1	6	2 894	1.02	2 953	1.03	2 424	1.02
2	3	3 052	1.03	2 842	1.02	2 813	1.02
2	4	3 524	1.04	3 742	1.04	3 294	1.03
2	6	3 250	1.02	3 134	1.02	3 042	1.02
3	3	2 535	1.03	2 464	1.04	2 246	1.04
3	4	3 245	1.04	3 143	1.03	2 943	1.03
3	6	3 024	1.04	2 943	1.02	2 453	1.02
4	3	4 023	1.02	3 032	1.02	2 942	1.03
4	4	5 242	1.05	4 987	1.04	3 853	1.03
4	6	4 622	1.04	4 342	1.02	3 945	1.02
5	4	4 524	1.02	4 245	1.02	4 013	1.02
5	5	4 324	1.05	4 142	1.05	3 542	1.05
5	6	4 242	1.04	4 023	1.03	4 042	1.02
6	4	5 043	1.06	3 952	1.03	2 343	1.03
6	5	5 153	1.06	3 592	1.03	3 029	1.02
6	6	5 353	1.05	4 023	1.03	2 592	1.03

从表 2 中可以看出，3 种方法在负载均衡系数相同的情况下达到基本一致的负载均衡度，但文中方法较经典的 Kmetis

方法以及文献 [13] 方法更为均衡，同时文中方法在划分出与经典的 Kmetis 方法以及文献 [13] 方法相同的区域数时，平均通信量仅为 2 998 MByte，较经典的 Kmetis 方法以及文献 [13] 方法分别低 22.6% 和 13.9%，这是因为文中方法在对经典的图划分模型进行改进的同时，采用改进的量子遗传算法对目标函数进行寻优，因此，能在保证具有最小通信开销的同时保持较优的负载均衡度。

### 5 结论

为了获得高效的并行计算性能，需要在各处理器上合理地分配任务，为此，设计了一种基于图划分和量子遗传算法的并行计算模型。在传统的图规划的基础上通过将传统的无向图改为有向图、在速度慢的处理节点上分配较少的任务量和对不同的依赖任务赋予不同的权值等，从而获得改进的图划分模型，并定义了以最小化通信开销和保持负载均衡度为目标的适应度函数。最后，定义了一种改进的量子遗传算法实现对适应度函数求解。仿真实验表明，文中方法能有效地实现高效并行计算，较其它方法具有较大的优越性。

#### 参考文献：

[1] 马永刚, 谭国真, 杨际祥, 等. 一种改进的并行计算图划分模型 [J]. 小型微型计算机系统 [J]. 2011, 32 (3): 416-420.

[2] Kc K, Anyanwu K. Scheduling Hadoop Jobs to meet deadlines [A]. Cloudcon' 10 Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science [C]. 2010: 388-392.

[3] 陈国良, 孙广中, 徐云, 等. 并行计算的一体化研究现状与发展趋势 [J]. 中国科学, 2009, 54 (8): 1043-1049.

[4] Polo J, Carrera D, Becerra Y, et al. Performance-driven task co-scheduling for map-reduce environment [A]. 12 th IEEE/IFIP Network Operation and Management Symposium [C]. 2010: 373-380.

[5] 万军, 赵不贻. 并行计算的 Petri 网建模和 FPGA 实现 [J]. 计算机应用研究, 2013, 30 (9): 2660-2663.

[6] 张雪萍, 龚康莉, 赵广才. 基于 MapReduce 的 K-Medoids 并行算法 [J]. 计算机应用, 2013, 33 (4): 1023-1025, 1035.

[7] 李成华, 张新访, 金海, 等. MapReduce: 新型的分布式并行计算编程模 [J]. 计算机工程与科学, 2011, 33 (3): 129-135.

[8] 杨燕, 王全根, 黄波. 蚁群聚类算法的并行化设计与实现 [J]. 控制工程, 2013, 20 (3): 411-414.

[9] 于方, 郑晓薇, 孙晓鹏. 基于多层 k 路划分的三维网格并行任务分配策略 [J]. 计算机工程与设计, 2010, 31 (2): 287-294.

[10] Grady L, Schwartz E L. Isoperimetric graph partitioning for image segmentation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2006, 28 (3): 469-475.

[11] Hendrickson B, Kolda T G. Graph partitioning models for parallel computing [J]. Parallel Computing, 2000, 26: 1519-1534.

[12] Walshaw C. The graph partitioning archive [EB/OL]. <http://Staffweb.cms.gre.ac.uk/~c.walshaw/partition/>. 2009.

[13] 马永刚, 谭国真, 王启冬. 基于 0-1 规划的并行计算图划分模型 [J]. 小型微型计算机系统, 2011, 32 (8): 1613-161.