

多核集群任务分配问题的 0-1 整数规划求解模型^①

杨际祥^② 凌 玲

(重庆交通大学数学与统计学院 重庆 400074)

摘 要 研究了典型多核集群任务分配中的节点内通讯特性。基于 0-1 整数非线性规划模型和线性松弛技术,给出了一种 0-1 整数线性规划任务分配问题求解优化模型。由于节点内的通讯量与通讯延迟较大,以最小化计算代价和节点间通讯代价为研究目标的传统求解模型具有严重的局限性,而该求解模型考虑了节点内通讯代价,并采用了线性规划松弛技术,其目标是最小化计算代价、节点间通讯代价和节点内通讯代价。计算结果验证了提出的模型的有效性。

关键词 多核集群,任务分配问题(TAP),0-1 整数规划,线性规划松弛

0 引 言

任务分配问题(task allocation problem, TAP)是并行计算领域的一个经典问题,任务分配旨在将一组任务分配到一组计算节点上,以最小化总成本。总成本通常包括执行成本和节点间(或处理器间)通讯成本。然而,随着基于多核节点的以层次方式构成的集群成为并行计算的一种发展趋势,通讯成本不但包括节点间通讯成本,还应包括节点内通讯成本,而节点内通讯成本又与节点内的通讯量和通讯延迟两个因素密切相关。文献[1,2]通过通讯量和通讯延迟展示了节点内通讯的重要性。文献[1]指出,网络附属存储(NAS)基准测试中超过 50% 的通讯量是通过节点内通讯完成的。文献[2]研究表明,在典型的多核集群系统中,在节点内的通讯延迟与节点间的通讯延迟之比随消息的增大而增大。因此,考虑并优化节点内通讯与优化节点间通讯成本同等重要。传统的任务分配问题(TAP)求解以最小化节点计算代价与节点之间的通讯代价为研究目标,具有严重的局限性。

现有的求解 TAP 的方法大致可分为三类,即图理论法、数学规划法以及启发式方法^[3]。数学规划方法将任务分配问题转化为优化问题,并使用数学规划技术^[3,4]进行求解。如果未知变量为整数,则该问题为整数规划问题。在实际情况中,通常未知变量个数有限,该问题为一 NP 难问题。0-1 整数规划为整数规划的一种特殊情况,变量或为 0 或为 1,该问题仍为一 NP 难问题。传统的四个或者更多个计算机上的任务分配问题的求解为一 NP 完全问题,更不必说新的多核集群任务分配问题(task allocation problem in multi-core clusters, TAPMC)的求解为一难解问题。本文使用整数规划理论求解 TAPMC,首先给出一个 0-1 整数非线性规划求解模型,然后借助线性规划松弛技术与优化方法给出一种新的 0-1 整数线性规划模型。

1 传统 TAP 的 0-1 整数规划模型

不失一般性,假设 n 个任务分配到 m 个计算节点上,部分任务间彼此存在通讯。如果任务 i 被分配到计算节点 j 上,产生的执行成本设为 e_{ij} ,而设当

① 国家自然科学基金(11401061),重庆市自然科学基金(KJ1400316)和交通运输部建设科技基金(2014318223030)资助项目。

② 男,1975 年生,博士,副教授;研究方向:并行计算与系统可靠性等;联系人,E-mail: jixiang_yang@126.com
(收稿日期:2015-12-16)

任务 i 与任务 k 未被分配到同一个计算节点上产生的通讯成本为 c_{ik} 。二元变量 x_{ij} 定义为 1, 当且仅当任务 i 分配到计算节点 j 上, 反之为 0。传统任务分配问题可转化为 0-1 整数规划问题, 该 0-1 整数规划问题旨在最小化完成任务的总成本, 如模型

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^m e_{ij} x_{ij} + \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m c_{ik} x_{ij} \\ & - \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m c_{ik} x_{ij} x_{kj} \\ \text{s. t.} \quad & \begin{cases} \sum_{j=1}^m x_{ij} = 1 \quad (i = 1, 2, \dots, n) \\ x_{ij} \in \{0, 1\} \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, m) \end{cases} \end{aligned} \quad (1)$$

所示。

0-1 整数规划问题可分为整数规划问题或者非线性规划问题, 部分研究者给出了该问题的求解方法。基于分枝限界技术, 文献[5]给出了一种求解该问题的方法。文献[6]使用 Lagrangian 对偶公式分枝限界法给出了一种有效的求解算法; 文献[7, 8]给出一种解决整数规划问题的列生成模型, 并且证明了它们的计算性能; 文献[4]引入一种混合整数规划模型, 当存在大量的任务通讯时尤其有效, 求解大规模问题时比其它精确算法更为有效。

2 多核集群任务分配问题 (TAPMC) 求解模型

2.1 0-1 整数非线性规划模型

0-1 整数规划模型(式(1))能够有效求解传统任务分配问题, 但无法有效求解多核集群任务分配问题。在本文中, 考虑了多核集群任务分配问题的 0-1 整数规划问题, 给出多核集群任务分配问题的一种 0-1 整数非线性规划求解模型, 如模型

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^m e_{ij} x_{ij} + \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} \bar{x}_{kj} c_{ik} \\ & + \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} x_{kj} I_{ik} \\ \text{s. t.} \quad & \begin{cases} \sum_{j=1}^m x_{ij} = 1 \quad (i = 1, 2, \dots, n) \\ x_{ij} \in \{0, 1\} \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, m) \end{cases} \end{aligned} \quad (2)$$

所示。

在模型式(2)中, 目标函数包括以下三个部分:

$$(1) \sum_{i=1}^n \sum_{j=1}^m e_{ij} x_{ij}, \text{ 表示总执行成本;}$$

$$(2) \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} \bar{x}_{kj} c_{ik}, \text{ 表示节点间发生的总的通讯成本, 且}$$

$$\begin{aligned} \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} \bar{x}_{kj} c_{ik} &= \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} (1 - x_{kj}) c_{ik} \\ &= \sum_{i=1}^{n-1} \sum_{k=i+1}^n c_{ik} - \sum_{i=1}^{n-1} \sum_{k=i+1}^n c_{ik} \sum_{j=1}^m x_{ij} x_{kj}; \end{aligned}$$

$$(3) \sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} x_{kj} I_{ik}, \text{ 表示在节点内发生的总的通讯成本。}$$

其中, I_{ik} 表示任务 i 与任务 k 分配到同一节点上产生的节点内通讯成本, 总的节点内通讯成本

$$\sum_{i=1}^{n-1} \sum_{k=i+1}^n \sum_{j=1}^m x_{ij} x_{kj} I_{ik} = \sum_{i=1}^{n-1} \sum_{k=i+1}^n I_{ik} \sum_{j=1}^m x_{ij} x_{kj} \circ$$

为有助于求解 TAPMC 问题, 模型式(2)可表示为

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^m e_{ij} x_{ij} + \sum_{i=1}^{n-1} \sum_{k=i+1}^n c_{ik} \\ & + \sum_{i=1}^{n-1} \sum_{k=i+1}^n (I_{ik} - c_{ik}) \sum_{j=1}^m x_{ij} x_{kj} \\ \text{s. t.} \quad & \begin{cases} \sum_{j=1}^m x_{ij} = 1 \quad (i = 1, 2, \dots, n) \\ x_{ij} \in \{0, 1\} \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, m) \end{cases} \end{aligned} \quad (3)$$

2.2 0-1 线性整数规划模型

在模型式(2)中, 当任务 i 与任务 k 被分配到同一计算节点 j 上时, $\sum_{j=1}^m x_{ij} x_{kj}$ 等于 1, 反之等于 0。用 y_{ikj} 替换模型式(2)中的 $x_{ij} \bar{x}_{kj}$, $y_{ikj} \leq x_{ij}$, $y_{ikj} \leq \bar{x}_{kj}$, 并且 $y_{ikj} \geq 0$ 。若 x_{ij} 或者 $\bar{x}_{kj}$ 为 0, 则 y_{ikj} 为 0。当 x_{ij} 与 $\bar{x}_{kj}$ 都为 1 时, 为最小化成本, y_{ikj} 必为 1。用 z_{ikj} 替换模型中的 $x_{ij} x_{kj}$, $z_{ikj} \geq x_{ij} + x_{kj} - 1$, 并且 $z_{ikj} \geq 0$ 。若 x_{ij} 与 x_{kj} 都为 1, 则 y_{ikj} 为 1, 反之, 为最小化成本, z_{ikj} 必为 0。因此, 模型式(3)可表示为

$$\min \quad \sum_{i=1}^n \sum_{j=1}^m e_{ij} x_{ij} + \sum_{i=1}^{n-1} \sum_{k=i+1}^n [c_{ik} + \sum_{j=1}^m (I_{ik} z_{ikj} - c_{ik} y_{ikj})]$$

s. t. {

$$\sum_{j=1}^m x_{ij} = 1 (i = 1, 2, \cdots, n)$$

$$y_{ikj} \leq x_{ij} (i = 1, 2, \cdots, n-1; k = i+1, \cdots, n; c_{ik} > 0; j = 1, 2, \cdots, m)$$

$$y_{ikj} \leq \bar{x}_{kj} (i = 1, 2, \cdots, n-1; k = i+1, \cdots, n; c_{ik} > 0; j = 1, 2, \cdots, m)$$

$$z_{ikj} \geq x_{ij} + x_{kj} - 1 (i = 1, 2, \cdots, n-1; k = i+1, \cdots, n; I_{ik} > 0; j = 1, 2, \cdots, m)$$

$$y_{ikj} \geq 0 (i = 1, 2, \cdots, n-1; k = i+1, \cdots, n; c_{ik} > 0; j = 1, 2, \cdots, m)$$

$$z_{ikj} \geq 0 (i = 1, 2, \cdots, n-1; k = i+1, \cdots, n; I_{ik} > 0; j = 1, 2, \cdots, m)$$

$$x_{ij} \in \{0, 1\} (i = 1, 2, \cdots, n; j = 1, 2, \cdots, m)$$

(4)

进一步,引入变量 u_{ij} 与 v_{ij} 。若任务 i 未分配到计算节点 j , 则 u_{ij} (或者 v_{ij}) 为 0, 否则表示任务 i 与其它任务 (任务 $i+1, i+2, \cdots, n$) 之间发生的节点间 (或者节点内) 通讯总成本。因此, 模型式 (4) 可表示为

min

$$\sum_{i=1}^n \sum_{j=1}^m e_{ij} x_{ij} + \sum_{i=1}^{n-1} \sum_{j=1}^m (u_{ij} + v_{ij})$$

s. t. {

$$\sum_{j=1}^m x_{ij} = 1 (i = 1, 2, \cdots, n)$$

$$u_{ij} \geq \sum_{k=i+1}^n c_{ik} (x_{ij} - x_{kj})$$

$$(i = 1, 2, \cdots, n-1; j = 1, 2, \cdots, m)$$

$$v_{ij} \geq \sum_{k=i+1}^n I_{ik} (x_{ij} + x_{kj} - 1)$$

$$(i = 1, 2, \cdots, n-1; j = 1, 2, \cdots, m)$$

$$u_{ij} \geq 0 (i = 1, 2, \cdots, n-1; j = 1, 2, \cdots, m)$$

$$v_{ij} \geq 0 (i = 1, 2, \cdots, n-1; j = 1, 2, \cdots, m)$$

$$x_{ij} \in \{0, 1\} (i = 1, 2, \cdots, n; j = 1, 2, \cdots, m)$$

(5)

2.3 应用极大独立集优化 TAPMC 求解模型

尽管给出了求解 TAPMC 的 0-1 整数线性规划模型, 该模型仍需进一步优化。在计算节点内与节点间的通讯成本时, 假定通讯图为一完全图, 因此可能产生一些不必要的计算。为避免这些不必要的计算, 引入极大独立集理论以优化通讯成本的计算。

定义 1: 对于一个无向图 $G = (V, E)$ 及子集 $S (S \subset V)$, 若: (1) S 中的任意两个节点不相邻, 则称 S 为 G 的一个独立集; (2) S 为 G 的一个独立集, 且对任意 $v \in V - S$, 在 S 中至少存在一个节点邻接 v , 则称 S 为 G 的一个极大独立集。

极大独立集问题为 NP 难问题。事实上, 没有必要寻找极大独立集, 若能够通过启发式算法寻找到一个不错的独立集, 则可认为得到一个理想的结果, 且该独立集也可能是一个极大独立集。在图 1 中, 通讯图为一无向图, 且通讯矩阵为对称阵, 此处

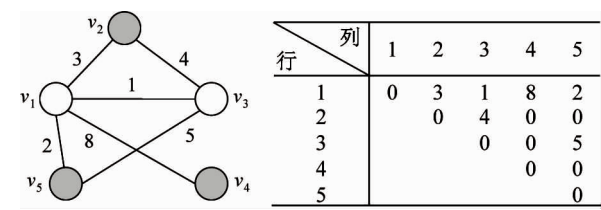


图 1 调整前的通讯图与通讯矩阵

算法 1: Adjust_RaC_Data

begin

Data: S (独立集: $\{1, \dots, u\}$)

Result: S (调整后的独立集)

按任务顺序对 S 中的所有任务排序;

$i_sfront \leftarrow 1, i_srail \leftarrow u, i_vrail \leftarrow n;$

while ($S[i_sfront] \leq n-u$)

if $S[i_srail] < i_vrail$

swap($S[i_sfront], i_vrail$);

$i_sfront++;$

$i_vrail--;$

else

$i_srail--;$

$i_vrail--;$

end

列 \ 行	1	2	3	4	5
1	0	1	3	8	2
2		0	4	0	5
3			0	0	0
4				0	0
5					0

图 2 调整后的通讯图与通讯矩阵

— 346 —

由一个上三角矩阵来表示。对于图 1 中的通讯图, $S = \{v_2, v_4, v_5\}$ 为一独立集。为优化矩阵计算, 给出一算法以调整通讯矩阵的行列数据, 如算法 1 所示。经调整后的通讯图与通讯矩阵如图 2 所示。通过观察可知, 为了计算通讯成本, 只需要计算

$$\sum_{i=1}^2 \sum_{k=i+1}^5 c_{ik} \sum_{j=1}^m y_{ikj} - \sum_{i=1}^2 \sum_{k=i+1}^5 c_{ik}, \text{ 而不必计算 } \sum_{i=1}^4 \sum_{k=i+1}^5 c_{ik} \sum_{j=1}^m y_{ikj} - \sum_{i=1}^4 \sum_{k=i+1}^5 c_{ik}, \text{ 从而降低了计算量。}$$

在算法 1 中, $swap(i, j)$ 方法完成第 i 行与第 j 行之间的数据交换操作, 同时完成第 i 列与第 j 列之间的数据交换操作。

3 计算结果

任务间的通讯使得分配决策比较困难。传统的任务分配问题的解决方法在任务间存在较少通讯时比较有效, 而在解决涉及大量的任务间通讯(如通讯密度为 50%、75% 或 100%)情形时显得较为困

难。文献[4]引入了一种任务分配问题整数求解模型 U_2 , 且计算结果表明其在当所求解问题存在大量的任务间通讯成本时效果较好。因此, 通过与 U_2 计算结果比较, 以研究本文模型的有效性。

计算节点具体环境为 Intel(R) Core(TM) 2 Quad CPU Q6000 @ 2.40GHz, 8G 内存, L2 Cache 4M, L1 Cache 32KB; Windows Server 2003; CPLEX 9.0^[9]。任务数 n 配置为 100, 150, 200; 计算节点数 m 配置为 4, 6, 8, 10; 通讯密度配置为 50%, 75%, 100%。

计算结果如图 3 ~ 图 5 所示。在任务数 $n = 100, 150, 200$ 时, 本文模型的求解时间皆快于 U_2 , 且随任务间通讯密度的增大, 本文模型的优势更加明显。此外, 当 $n = 150, m = 10$, 通讯密度 = 100%; $n = 200, m = 8, 10$, 通讯密度 = 50%, 75%; $n = 200, m = 6, 8, 10$, 通讯密度 = 100% 时, 模型 U_2 计算效率低下(超时), 而本文模型则表现出了良好的计算效率与可扩展性。

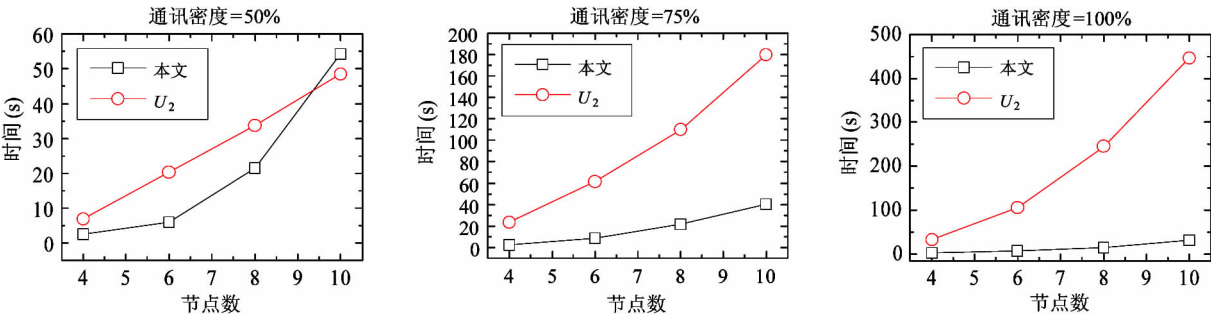


图 3 不同通讯密度下的任务(任务数 $n = 100$)求解时间随节点数的变化关系

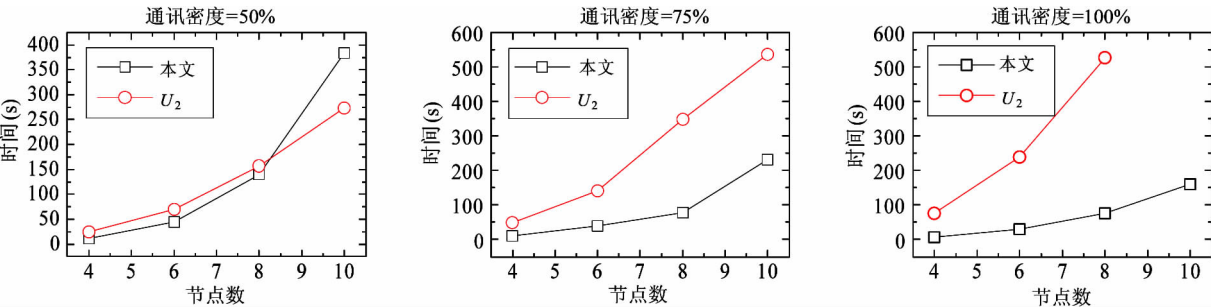


图 4 不同通讯密度下的任务(任务数 $n = 150$)求解时间随节点数的变化关系

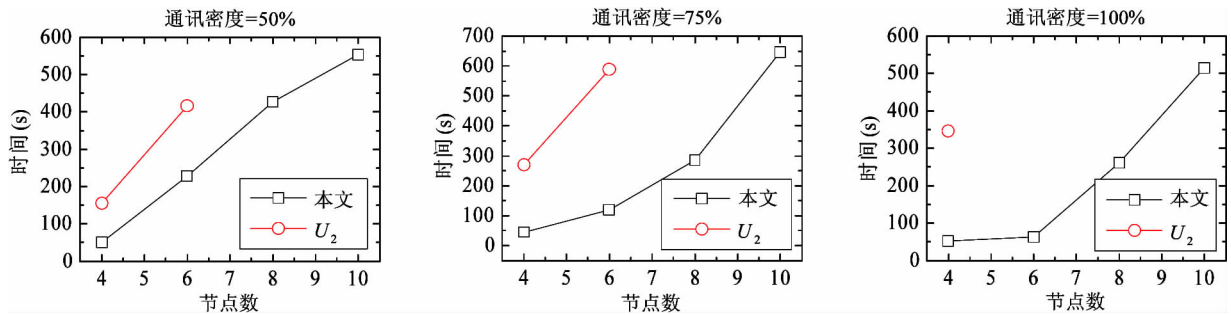


图 5 不同通讯密度下的任务 (任务数 $n=200$) 求解时间随节点数的变化关系

4 结 论

任务分配是多核集群系统中的一个重要问题。传统 TAP 为 NP 难问题,新的 TAPMC 的求解更具有挑战性。本文给出一种求解 TAPMC 的 0-1 整数非线性规划模型,并给出一种 0-1 整数线性规划模型及其优化方法。计算结果表明,当存在大量任务间通讯时,本文给出的计算模型比较有效。

参考文献

- [1] Chai L, Gao Q, Panda D K. Understanding the impact of multi-core architecture in cluster computing: a case study with Intel dual-core system. In: Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid, Rio De Janeiro, Brazil, 2007. 471-478
- [2] 涂碧波,邹铭,詹剑锋等. 多核处理器机群 Memory 层次化并行计算模型研究. 计算机学报, 2008, 31(11): 1948-1955
- [3] Ernst A, Jiang H, Krishnamoorthy M. Exact solutions to

task allocation problems. *Management Science*, 2006, 52(10): 1634-1646

- [4] Menon S. Effective reformulations for task allocation in distributed systems with a large number of communicating tasks. *IEEE Transactions on Knowledge and Data Engineering*, 2004, 16(12): 1497-1508
- [5] Soylu B. Heuristic approaches for biobjective mixed 0-1 integer linear programming problems. *European Journal of Operational Research*, 2015, 245(3): 690-703
- [6] Sherali H D, Smith J C. Higher-level RLT or disjunctive cuts based on a partial enumeration strategy for 0-1 mixed-integer programs. *Optimization Letters*, 2012, 6(1): 127-139
- [7] Alfandari L, Sadki J, Plateau A. Hybrid column generation for large-size covering integer programs: application to transportation planning. *Computers & Operations Research*, 2013, 40(8): 1938-1946
- [8] Rönnberg E, Larsson T. All-integer column generation for set partitioning: basic principles and extensions. *European Journal of Operational Research*, 2014, 233(3): 529-538
- [9] ILOG, Inc. ILOG CPLEX 9.0 User's Manual. 2003, <http://eaton.math.rpi.edu/cplex90html/pdf/usreplex.pdf>

A model for solving task allocation problems in multi-core clusters using 0-1 integer programming

Yang Jixiang, Ling Ling

(School of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing 400074)

Abstract

The in-node communication characteristics of the task allocation of multi-core clusters was studied. An optimized model for solving task allocation problems in multi-core clusters using 0-1 integer linear programming was proposed based on the 0-1 integer linear programming model and the linear relaxation technique. Considering that the traffic and delay of in-node communications are bigger, which brings serious limitations to the traditional models aiming to minimize the computing cost and the cost of the node-node communications, the proposed model takes account of the cost of in-node communications and adopts the technique of linear programming relaxation, with the aim to minimize the computing cost, the cost of node-node communications, and the in-node communication cost. The effectiveness of the model is verified by the computation result.

Key words: multi-core cluster, task allocation problem, 0-1 integer programming, linear programming relaxation