第 33 卷 第 4 期
1999 年 4 月

西 安 交 通 大 学 学 报
JOURNALOFXI'ANJIAOTONGUNIVERSITY

Vol.33 №4
Apr.1999

多处理器系统最优任务分配问题的一个近似算法

徐寅峰[1], 叶继昌[1], 迟学斌[2]
(1. 西安交通大学, 710049, 西安; 2. 中国科学院软件研究所)

摘要: 多处理器系统上的最优任务分配的研究是有效利用系统资源处理实际问题的热点课题,文中在考虑任务可分和任务不可分的两种多处理器最优任务分配问题上,首先提出了这两个问题在处理器的个数大于 1 时都是 NP-完全问题,其次给出了一个有效的近似算法,并证明了该算法所产生的解与最优解的近似比小于 2.

关键词: 多处理器;最优任务分配;计算复杂性;近似算法
中国图书资料分类法分类号: TB114.4

AnApproximateAlgorithmforOptimizingTaskAssignment inaMultiprocessorSystem

Xu Yinfeng[1], *Ye Jichang*[1], *Chi Xuebin*[2]
(1.Xi'anJiaotongUniversity,Xi'an710049,China;2.InstituteofSoftware,AcademiaSinica)

Abstract: Theseparableandnonseparableproblemassociatedwithamultiprocessorsystemisstudiedforoptimizingtaskassignment.BothproblemsareshowntotheNP-hardwhenthenumberofprocessorsisgreaterthan one.Theapproximateratiooftheproposedalgorithmislessthantwo.

Keywords: *multiprocessorsystem; optimal task assignment; computational complexity; approximate algorithm*

多处理器分布式系统上的任务分配问题是指:在分布式计算环境下,如何最有效地利用系统资源来完成一组有限的任务集合. 这方面的研究结果在大规模数值计算、VLSI 和计算机网络技术方面都有很好的应用背景. 在理论方面,由于许多最优任务分配问题为 NP-完全问题[1,2],所以如何构造有效启发式算法或近似算法是目前研究的热点领域[3,4].

以往研究的最优任务分配问题限定给定的任务集合 $T=\{t_1, t_2, \cdots, t_k\}$ 在一个含有 n 个处理器 $P=\{p_1, p_2, \cdots, p_n\}$ 的分布式系统上完成,同时假设每项任务在不同的处理器上完成具有不同的费用,并且不同的处理器存在着通讯开销. 另外一种广泛加以研究的最优任务分配问题还限定给定的任务之

收稿日期: 1998-04-01. 作者简介: 徐寅峰,男 1960 年 9 月生,管理学院战略与决策研究所,教授.
基金项目: 国家自然科学基金重点资助项目(19731001).

间存在一种偏序关系. 以上两种处理器分布式系统上的最优任务分配问题的目标函数是完成任务集合的总费用[3~5]. 文献[3]应用图论中的匹配方法研究了极大极小准则下的一类分布式系统上的最优任务分配问题,问题的限定条件与文献[4,6]中对问题的限定相一致.

对于分布式计算系统上的任务分配问题的研究方法大致可分为 3 类:即图论方法,数学规划,及启发式算法. 由于大多任务分配问题为 NP-完全问题或 NP-难问题[1],以上几种方法所给出的求解算法多数是通过实际的计算数值比较来评价算法的优劣,所以难以给出有效求解算法所给出的解的近似比.

本文主要研究如下多处理器分布式系统上的任务分配问题. 给定一个任务集合 $T=\{t_1,t_2,\cdots,t_k\}$, n 个处理器 $P=\{p_1,p_2,\cdots,p_n\}$, 假定处理器之间的通讯费用不计,那么如何分配任务集合 T 到处理器集合 P 上,使得处理器所用的最大时间最小,我们考虑如下两类限定条件:

(1) 每个任务 t_i 只能在一个处理器上完成,一个处理器可以处理多个任务;

(2) 每个任务 t_i 可以分别在多个处理器上同时执行,但是如果 t_i 在 $P^*\subset P$ 上完成且 $|P^*|>1$,那么不能把其它的任务 $t_j(i\neq j)$ 分配给 P^* 中的处理器,同时一个处理器可以承担多个不同的完整任务.

我们首先指出如上两类问题在 $|P|\geq 2$ 时都是 NP-完全问题,然后给出一个有效的算法,并证明该算法所给出的解与最优解的近似比小于 2,最后给出有关的应用与实例.

1　问题的表示

令 $T=\{t_1,t_2,\cdots,t_k\}$ 为所要完成的任务集合,其中 t_i 表示完成第 i 个任务所需的时间. n 个处理器为 $P=\{p_1,p_2,\cdots,p_n\}$. 令 A 为一可行的任务分配方案,$t_p(A)$ 表示第 p 个处理器完成方案 A 所分配到该处理器上所有任务的总时间,那么我们所要求得的最佳分配方案的目标函数为:

$$t(A_0)=\min_{A}\max_{P} t_p(A) \qquad (1)$$

此目标函数表示用最少的时间在 P 上完成任务 T.

对于不同的限定条件,式(1)所规定的目标函数的最优解是不同的. 我们主要考虑如下两种限制条件:(Ⅰ) $\forall t_i\in T$ 只能在一个处理器上完成,一个处理器可以处理多个任务;(Ⅱ) $\forall t_i\in T$ 可以在多个处理器上同时执行,但是如果 t_i 在多于一个处理器上完成,那么对 t_i 进行处理的处理器就不能再对其它任务进行处理了,同时一个处理器可以承担多个不同的完整任务.

对于(Ⅰ)限制条件下的目标函数式(1)的最优方案设计问题称为不可分任务最优分配问题,简记为 NSOTA(Non-SeparableOptimalTaskAssignment);对于(Ⅱ)限制条件下的目标函数式(1)的最优方案设计问题称为可分任务最优分配问题,简记为 SOTA(SeparableOptimalTaskAssignment). 在一些特殊情况下,NSOTA与 SOTA具有等价性. 为了讨论方便起见,将这两个问题分开加以处理.

2　问题的计算复杂性

文献[1]中列举了许多 NP-完全的最优任务分配问题,由于我们所讨论的问题 NSOTA与 SOTA的目标函数为一个极大极小问题,且问题的约束条件不具有以往的图论结构[7,8],所以关于 NSOTA与 SOTA的计算复杂性应从另外一个方面来加以考虑. 我们借助如下 NP-完全问题来说明 NSOTA与 SOTA都是 NP-完全的.

集合分拆问题:给定有限集合 A 及 A 上的权函数 $c(a)\in Z^+$, $a\in A$,是否存在子集 $A'\subseteq A$ 满足 $\sum_{a\in A'}c(a)=\sum_{a\in A-A'}c(a)$?

集合分拆问题是一个 NP-完全问题(参见文献[1]). 对于 NSOTA与 SOTA,显然都是 NP 问题. 若我们仅考虑处理器的个数为 2 的情况,那么任务的个数大于 1 时 NSOTA与 SOTA等价. 实际上 NSOTA与 SOTA就是要寻求一种分配方案将任务集合 $T=\{t_1,t_2,\cdots,t_k\}$ 尽可能平均地分配到两个处理器上,即将 T 分为 T_1、T_2,使得 $T_1\cup T_2=T$,同时使得 $|\sum_{t_i\in T_1}t_i-\sum_{t_j\in T_2}t_j|$ 最小. 于是集合分拆问题就是 NSOTA与 SOTA的一个特殊情况. 若 NSOTA与 SOTA有多项式算法,则集合分拆问题也是多项式时间可解. 于是我们有如下结论:

定理 1　NSOTA与 SOTA都是 NP-完全问题.

集合分拆问题,应用动态规划可以给出其解的伪多项式算法,对于这样的问题通常可能设计出有效的近似算法. 下一节中我们将讨论 NSOTA与 SOTA的近似算法.

3 有效近似算法设计

由上一节的讨论可知,NSOTA与SOTA都是NP-完全问题.如何给出NP-完全问题的有效近似算法,则是实际处理NP-完全问题的一个核心问题.对于NSOTA与SOTA,我们分别给出如下算法:

算法1 (NSOTA)

输入:k 个任务 $T=\{t_1,t_2,\cdots,t_k\}$,处理器个数 n.

输出:k 个任务分配到 n 个处理器上的一种方案.

步1:将 k 个任务全排序变为 $t'_1\geq t'_2\geq\cdots\geq t'_k$.

步2:求 $\sum_{i=1}^{k}t'_i=c$.

步3:如果 $t'_k\geq c/n$,则去掉 t'_k,$n=n-1$,$c=c-t'_k$;否则 $t'_k+t'_{k-1}=t'_{k-1}$,去掉 t'_k.对 $k=k-1$ 重复步3.

算法2 (SOTA)

输入:k 个任务 $T=\{t_1,t_2,\cdots,t_k\}$,处理器个数 n.

输出:k 个任务分配到 n 个处理器上的一种方案.

步1:将 k 个任务全排序变为 $t'_1\geq t'_2\geq\cdots\geq t'_k$.

步2:求 $\sum_{i=1}^{k}t'_i=c$.

步3:如果 $t'_k\geq c/n$,计算 $\lceil nt'_k/c\rfloor=l$,将 t'_k 按 t'_k/l 分给 l 个处理器,去掉 t'_k,$n=n-l$,$c=c-t'_k$;否则 $t'_k+t'_{k-1}=t'_{k-1}$,去掉 t'_k.对 $k=k-1$ 重复步3.

根据以上算法的运算可以得出如下结论:

定理2 算法1、算法2的时间复杂性为 $O(k\lg k)$,空间复杂性为 $O(k)$.

证明 算法1、算法2的步1可在 $O(k\lg k)$ 时间内完成,步2、步3可用 $O(k)$ 时间完成.于是总的运算量为 $O(k\lg k)$.

由定理2可知算法1、算法2在计算时间上是非常有效的.

4 近似比

令 $p(1),p(2),\cdots,p(n)$ 为算法1给出的任务 T 对应于NSOTA的解,$p'(1),p'(2),\cdots,p'(n)$ 为算法2给出的任务 T 对应于SOTA的解,这里 $p(i)$ 和 $p'(i)$ 表示第 i 个处理器上所分配的任务的时间总和.

$$\left.\begin{aligned}C_1(T,n)&=\max_{1\leq i\leq n}p(i)\\C_2(T,n)&=\max_{1\leq i\leq n}p'(i)\end{aligned}\right\}\qquad(2)$$

$C'_{opt}(T,n)$ 为NSOTA的最优解,$C''_{opt}(T,n)$ 为STOA的最优解,即

$$\left.\begin{aligned}C'_{opt}(T,n)&=\min_{A\in N}\max_{p}t_p(A)\\C''_{opt}(T,n)&=\min_{A\in S}\max_{p}t_p(A)\end{aligned}\right\}\qquad(3)$$

其中 N 为求解NSOTA的所有算法集合;S 为求解SOTA的所有算法集合.

定理3

$$\left.\begin{aligned}C_1(T,n)&<2C'_{opt}(T,n)\\C_2(T,n)&<2C''_{opt}(T,n)\end{aligned}\right\}\qquad(4)$$

证明 以下仅证明定理中的第2个不等式,第1个不等式可以通过类似的证明方法得到.对于问题SOTA,由于将 $\sum_{i=1}^{k}t_i=c$ 总任务安排到 n 个处理器上完成,所以最优解 $C''_{opt}(T,n)$ 的一个显然下界为 $\frac{1}{n}\sum_{i=1}^{k}t_i$,即

$$C''_{opt}(T,n)\geq\frac{1}{n}\sum_{i=1}^{k}t_i\qquad(5)$$

根据算法2,我们分不同情况来考虑 $C_2(T,n)$.

(1) 若 $t'_k\geq c/n$,算法2将任务 t'_k 分配给 $\lceil nt'_k/c\rfloor=l$ 个处理器,在 l 个处理器上所处理的任务都是 t'_k/l,且有

$$2c/n>t'_k/l\geq c/n\qquad(6)$$

即在这 l 个处理器上的费用小于 $2C''_{opt}(T,n)$,对于剩余的费用有如下不等式

$$\sum_{i=1}^{k-1}t_i/(k-l)\leq c/n\qquad(7)$$

于是,要考虑其它 $n-l$ 个处理器上的任务安排.在这一情形下,每个处理器上的费用同样不会超过 $2C''_{opt}(T,n)$.

(2) 如果 $t'_k<c/n$,由于已对任务 T 进行了重排序,所以有 $k>n$ 且 $t_i<c/n$,$1\leq i\leq k$.由算法2,将 t_{k-1} 加到 t_k 上,直到 $c/n\leq\sum_{j=1}^{i}t_{k-j+1}$.$c/n>\sum_{j=1}^{i-1}t_{k-j+1}$ 时,将 $t_k,t_{k-1},\cdots,t_{k-i}$ 放到同一个处理上来完成.在这种情况如下不等式仍然成立

$$\sum_{j=1}^{i}t_{k-j+1}<2c/n\leq 2C''_{opt}(T,n)\qquad(8)$$

且

$$\sum_{j=1}^{k-i}t_j/(k-1)\leq c/n\qquad(9)$$

于是,其它 $n-1$ 个处理器上的任务安排同样不会超过 $2C''_{opt}(T,n)$.由以上讨论知,定理3的结论成立.

5 实例与应用

如下仅对限定条件(Ⅰ)给出一个实例以及在限

定条件(Ⅱ)上的一个应用.

对于 T = {99,99,9,11}, n =3 ,求 NSOTA的最优解.其最优解显然为 p_1 处理问题 t_1, p_2 处理问题 t_2, p_3 处理问题 t_3 和 t_4,完成 T 所用时间为 102. 应用算法 1,产生的分配方案是 p_1 处理问题 t_1 和 t_2, p_2 处理问题 t_3, p_3 处理问题 t_4,完成 T 所用时间为 198. 算法 1 产生结果的近似比为 198/102.

考虑用反迭代法并行求解 $m \times m$ 对称三对角矩阵 T 的特征向量,这里假定特征值已经求出. T 有如下形式

$$T = \begin{bmatrix} d_1 & e_2 & & \\ e_2 & d_2 & \ddots & \\ & \ddots & \ddots & e_{m-1} \\ & & e_m & d_m \end{bmatrix} \qquad (10)$$

进一步假设已有在 n 个处理器上求解 m 阶问题的并行计算方法 t. 根据求此问题的特点,如果 T 的某个非对角线元素是 0,则此 m 阶问题可化成两个小规模问题来求解. 这样做既可以减少计算时间又可以得到准确的特征向量[10]. 因此,如果有多个非对角线元素是 0,我们就可以得到一系列的子问题. 为了减少在进行正交化过程中的计算和通讯开销,我们就要对子问题的任务分配进行合理的安排,这就是本文所考虑的限制条件 Ⅱ的问题,利用此方法已在求解特征值问题中得到了很好的应用.

参考文献:

[1] GareyM,JohnsonD.Computersandintractability,aguide tothetheoryofNP-completeness.SanFrancisco:Freeman, 1979.

[2] LoVM.Heuristicalgorithmsfortaskassignmentindistributedsystem.IEEETransComputer,1988,37 (11):1384～1397.

[3] ShenC,TsaiW.Agraphmatchingapproachtooptimaltask assignmentindistributedcomputingsystemusingaminimax criterion.IEEETransComputer,1975,34 (3):197～203.

[4] WuW.Onruntimeparallelschedulingforprocessorloadbalancing.IEEETransonParallelandDistributedSystem, 1997,8 (2):173～185.

[5] LeeC,ShinK.Optimaltaskassignmentinhomogeneousnetworks.IEEETransonParallelandDistributedSystem. 1997,8 (2):119～129.

[6] RamamoorthyC.Optimalschedulingstrategiesinamultiprocessorsystem.IEEETransComputer,1972,21 (2):137～146.

[7] ScholzP,HarbeckE.Taskassignmentfordistributedcomputing.In:AdyancesinParallelandDistributedComputing. Washington:IEEEComputerSocietyPress,1997.270～277.

[8] AhmadI,KafilM.Aparallelalgorithmforoptimaltaskassignmentindistributedsystem.In:AdvancesinPralleland DistributedComputing.Washington:IEEEComputerSociety Press,1997.284～290.

[9] SunH.Newboundsontimeandnumberofprocessorsfor multiprocessoroptimalschedules.WuhanUniversityJNatural Sci,1996,1 (3/4):350～355.

[10]ChiXuebin.Parallelsolverofgeneralizedeigenproblemon dawning1000.In:AdvancesinParallelanddistributedcomputing.Washington:IEEEComputerSocietyPress,1997. 144～148.

(编辑　赵大良)

(上接第 97 页)

1,事实上 1- α～O(kυ),又由式(5),我们可断定在不太长的时间内 β(m)～O($\frac{1}{n}$).

参考文献:

[1] FoiasC,SellGR,TemamR.Inertialmanifoldsfornonlinearevolutionaryequations.JournalofDifferentialEquations, 1988,73:309～353.

[2] FoiasC,ManleyO,TemamR.Ontheinteractionofsmall eddiesintwo-dimensionalturbulenceflows.MathModeling andNumericalAnalysis,1988,22:93～114.

[3]TemamR.Navier-Stokesequationandnonlinearfunctional analysis.SIAM,Philadelphia,1983.

[4] 李开泰,马逸尘. 数理方程 HILBERT空间方法(下). 西安:西安交通大学出版社,1992.

[5] DebusscheA,TemamR.Convergentfamilyofapproximate inertialmanifolds.JMathPuresAppl,1994,73:489～522.

[6]MarionM,TemamR.NonlinearGalerkinmethods.SIAMJ NumerAnal,1989,26:1139～1157.

[7]LiKaitai,HouYanren.FouriernonlinearGalerkinmethodfor Navier-Stokesequations.DiscreteandContinuousDynamical Systems,1996,2 (4):479～524.

(编辑　杜秀杰)