一种多DAG任务共享异构资源调度的费用优化方法

田国忠[1,2],肖创柏[1],谢军奇[1]
(1.北京工业大学计算机学院,北京 100124;2.新疆工程学院计算机工程系,新疆乌鲁木齐 830091)

摘 要: 针对多DAG(Directed Acyclic Graph)共享一组异构资源在调度吞吐量最大化基础上的费用优化问题,本文提出了一种基于总费用变化量探测的费用优化算法PDTC(based on the Probe of the Total Cost Decrease),目的在于尽可能降低有优化条件的多个DAG的总费用.实验表明,该算法不仅能使得各DAG充分利用期限内的冗余时间,也能够在一定程度上降低多个DAG调度执行的总费用.

关键词: 多DAG调度;紧急水平;相对于Deadline的满意度;规范化费用
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2014)09-1767-08
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.09.016

An Cost Optimization Methods for Scheduling Concurrent Multiple DAGs Sharing Heterogeneous Resources

TIAN Guo-zhong[1,2], XIAO Chuang-bai[1], XIE Jun-qi[1]
(1. *College of Computer Science and Technology, Beijing University of Technology, Beijing* 100124, *China*;
2. *Department of Computer Engineering, Xinjiang Institute of Engineering, Urumqi, Xinjiang* 830091, *China*)

Abstract: Aiming to the new problems of cost optimization for concurrent multiple DAG(Directed Acyclic Graph) workflow sharing heterogeneous resource on the basis of maximizing overall throughput of DAGs scheduling, we propose an algorithm called PDTC(based on the Probe of the Total Cost Decrease)to minimize the total cost of these DAG while meeting a user-defined deadline. Last experiments demonstrate that these algorithms and methods can improve the related performances.

Key words: multiple DAGs scheduling; urgency level; satisfaction degree to deadline; normalized cost

1 引言

目前很多的异构分布式系统,如效用网格或公有云计算等系统,资源提供商往往会基于租赁的销售模式以及基于使用量和性能指标的计费模式对用户应用所提供的计算服务进行计费[1].因此,为用户DAG(Directed Acyclic Graph,有向无环图)应用任务调度分配不同性能的资源,其DAG应用任务调度运行的费用是不同的,并且不同的调度算法所引起的费用也是不同的.为了尽可能降低DAG应用任务在系统上运行费用,会涉及到调度的费用优化问题.针对一些分布式计算系统下DAG调度的费用优化问题,相关的研究[2~6]提出了很多好的方法和解决方案,但这些关于DAG调度的方法还主要是针对单个DAG在一组资源上的调度及其费用优化,并且基本都有一个期限划分(deadline-assignment)的重要步骤:即根据提出的不同方法将整个DAG的期限约束Deadline分配到各任务,然后基于任务所分配的时间窗口进行费用优化.如文献[6]中针对具有期限约束的DAG任务在云计算环境下执行费用的优化问题,提出了利用云计算环境资源的"可按需扩展的机制",并采用Deadline-Assignment与EDF(Earliest Deadline First,最早期限优先)[7,8]方法相结合的方法解决具有期限约束的DAG费用优化问题.虽然该方案能够解决多个DAG的调度问题,但由于其解决方案是利用Amazon弹性云计算环境下资源可"按需扩展"的机制,根据新到达DAG随时申请新的资源组进行调度,这样就会有多个不同的资源组分别为不同的DAG单独提供服务,本质上仍然属于单DAG在一组资源上单独调度的费用优化问题,存在着资源利用率不高的问题.另外,其采用的Deadline-Assignment和EDF相结合的方法也并不适用于解决具有期限约束的多个DAG共享资源混合调度的费用优化问题[1].

收稿日期:2012-12-16;修回日期:2014-04-28;责任编辑:梅志强
基金项目:国家自然科学基金(No.61363004,No.61262088,No.61363083)

在有限的资源上和有期限约束的时间内,显然如果要让 DAG 的吞吐量越大,各个 DAG 的任务选择较为便宜资源(因而也较慢)的机会和可能性就越小.因此当多个具有期限约束的 DAG 共享一组资源进行调度时,DAG 的吞吐量最大化和费用最小化是一对矛盾,往往需要根据应用系统的具体需求在两个调度目标之间进行平衡,是个较为复杂的问题.然而,从系统管理的角度来看,吞吐量最大化常常是一个重要的目标,因此在 DAG 吞吐量最大化基础上降低所有 DAG 任务的费用是需要研究和解决的重要问题之一.本文所提出和要解决的问题是:对共享异构资源的一组具有期限约束的多个 DAG,利用吞吐量最大化算法进行预调度后,若仍存在有冗余时间,就具备了费用优化的条件,则有必要对这些 DAG 进行总费用优化.那么如何在吞吐量最大化的基础上,既要满足 DAG 组中的每个 DAG 都能在其期限之内完成的约束条件,又要利用每个 DAG 期限所留下的冗余时间而降低所有 DAG 总费用,是个较为复杂的问题.就我们现有知识范围来看,目前为止还未见有相关的研究和文献提出上述问题及其解决方法.为此,本文提出了一种基于总费用变化量探测的费用优化算法 PDTC(based on the Probe of the Total Cost Decrease).

2 多 DAG 调度的总费用优化

2.1 问题描述

假设有 n 个由用户指定了期限约束的 DAG(G_1, G_2,…, G_i,…, G_n)在一组有限数量 q 个异构分布式计算系统资源(M_1, M_2,…, M_i,…, M_q)上同时进行调度.其中,这组资源模型、DAG 任务图及图中任务的 $\text{rank}_u(n_i)$以及 DAG 的期限约束 $t_{\text{Deadline}-G_i}$等与文献[1,3,9]的有关假设、定义和计算方法一致.关于任务在资源上执行的经济花费,文献[2~5]均假设任务在资源上运行的时间越短,其经济花费也越高.在一些现实的网格或云计算环境中资源提供商,一般是按照所提供的资源类型和用户应用任务使用资源的时间进行计费,不同类型的资源都有一个单位时间的使用价格,性能越高的资源,价格也越高,如文献[6]中假设每个机器都有一个单位时间的使用价格,与现实云计算资源的计价模式基本相同.在本文,我们也类似地假设异构计算资源 M_i 依据其提供的计算服务及运算速度有一个单位时间使用价格 p_{M_i},并且任务在每个资源上的执行时间都不一样,在价格较高的资源上执行时间也较短.若某 DAG G_i 中的第 k 个任务表示为 G_{i-k},且在某资源上 M_i 的执行时间为 w_{G_{i-k},M_i},则任务 G_{i-k} 在资源 M_i 上经济花费应为 $p_{M_i}\cdot w_{G_{i-k},M_i}$.最终的调度目标是:在期限内完成的 DAG 吞吐量最大化的基础上,进一步降低这些 DAG 总的执行费用.

文献[1]中已经对 DAG 吞吐量最大化问题做了研究和分析,并提出了 MDRS 算法.该算法能够对 n 个 DAG 共享 q 个资源的混合调度中可能存在的"过饱和"情况进行探测和处理,并能使得这组 DAG 期限内完成的数量最大化.因此,如果确定了一组 n 个 DAG 中能在期限内完成的 DAG 最大数量为 n_c,并且这 n_c 个 DAG 的完成时间相对于各自的期限约束仍存在 DAG 有冗余时间,这意味着存在费用优化的可能,可将更便宜的机器资源分配给有关任务,尽可能降低这 n_c 个能在期限内完成的 DAG 的总费用,而费用优化需满足的约束条件是这 n_c 个 DAG 能够在期限内完成.

由于每个资源的单位时间使用价格和每个任务在每个机器资源上的估计执行时间确定,因此只要确定了 DAG 的所有任务在资源上的映射关系,就能够确定 DAG 在资源上的执行费用[10]和所有 DAG 调度执行的总费用.针对上述问题,本文提出了总费用优化 PDTC 算法.

2.2 总费用优化 PDTC 算法的主要步骤及有关符号表示

PDTC 算法的主要思想是在保证满足这些 DAG 在各自期限约束内完成的前提下,采用所有 DAG 总费用降低量探测的方法进行费用优化.在优化过程中利用 MDRS 算法来确定下一个要优化的任务 G_{i-k},这个任务可以在 MDRS 算法确定的 M_{HEFT}(由于 MDRS 算法在选择了某个任务后将用 HEFT 算法调度该任务,因此该资源记为 M_{HEFT})和 M_{HEFT}以外的资源中做出选择.当该任务选择了某个资源并用 MDRS 预调度所有 DAG 的剩余未被优化任务后,如果不会出现有 DAG 的完成时间超过其期限的情况,则计算 DAG 的总费用.与当前的最低总费用相比,在所有资源中,如果 G_{i-k} 选择了某个资源后的总费用降低量最大,则这个最低总费用被置为当前最低总费用,并且该资源被 G_{i-k} 选为优化后的资源.继续利用 MDRS 算法选取下一个优化任务按上述步骤优化,直到所有任务优化完毕.

假设最初的当前最低总费用为 $\text{Cost}_{\text{Optim}-\text{total}}$(初始值为吞吐量最大化算法 MDRS 调度方案中所有 DAG 的总花费,表示为 $\text{Cost}_{\text{MDRS}-\text{total}}$),当某任务 G_{i-k} 被确定为当前要优化的任务,这个任务在 MDRS 算法下的资源为 M_{HEFT},则其余资源作为任务 G_{i-k} 的可选资源放入集合 $M_{G_{i-k}}$ 中.如果任务 G_{i-k} 选择了 $M_{G_{i-k}}$ 中的某个资源 M_q($\forall M_q \in M_{G_{i-k}}$),则利用 MDRS 算法将剩余所有未被优化任务进行预调度.如果预调度后不会出现有 DAG 的完成时间超出期限约束,则计算任务 G_{i-k} 选择 M_q 情况下的所有 DAG 总花费,被表示为 $\text{Cost}(G_{i-k},M_q)_{\text{total}}$.而 $\text{DTC}(G_{i-k},M_q)=\text{Cost}_{\text{Optim}-\text{total}}-\text{Cost}(G_{i-k},M_q)_{\text{total}}$ 则表

示,相对于当前最低总花费,任务 G_{i-k}选择了 M_q 时所有 DAG 总花费的变化量.遍历 $M_{G_{i-k}}$所有资源,如果选择 $M_{G_{i-k}}$中某个资源 M_q 导致出现有 DAG 的完成时间超出其期限约束,则令 DTC(G_{i-k}, M_q) = 0.假如 $M_{G_{i-k}}$中存在使得 DTC(G_{i-k}, M_q) > 0 的 M_q 资源,则从 $M_{G_{i-k}}$中选择使得 DTC(G_{i-k}, M_q)最大的 M_q 作为 G_{i-k}优化后的资源,并将该 M_q 资源对应的 Cost (G_{i-k}, M_q)$_{\text{total}}$作为当前最低总费用,继续用 MDRS 的方法选取下一个要优化的任务按上述步骤优化,直到所有任务优化完毕.

2.3 完整 PDTC 算法

算法 1　PDTC

1:**输入**:在 MDRS 算法下能够在期限内完成的 n_c 个 G_i 放入待调度优化 DAG 集合 $G_{\text{unoptimized}}$;
2:每个 DAG 的 $t_{\text{Deadline}-G_i}$;一组可用资源 R;
3:每个 G_i 的所有任务按向上权值 $\text{rank}_u(n_i)$值从大到小排序放入相应 G_i 的未优化任务集合 $G_{i-\text{unoptimized}}$;
4:每个 G_i 的已优化任务集合 $G_{i-\text{optimized}} \leftarrow \Phi$;　/* 对 $G_{i-\text{optimized}}$置空,Φ 代表 Null */
5:$\text{Cost}_{\text{Optim}-\text{total}} \leftarrow \text{Cost}_{\text{MDRS}-\text{total}}$;　/* $\text{Cost}_{\text{MDRS}-\text{total}}$为吞吐量最大化 MDRS 算法调度后的所有 DAG 的总费用 */
6:**输出**:费用优化后的 n_c 个 DAG 的调度方案(存储在 $G_{i-\text{optimized}}$中)
7:Procedure PDTC($G_{\text{unoptimized}}$, $t_{\text{Deadline}-G_i}$, R, $G_{i-\text{unoptimized}}$, $G_{i-\text{optimized}}$)
8:while($G_{\text{unoptimized}} \neq \Phi$)do
9:　在已优化资源占据了资源的基础上,利用 $\text{MDRS}_{\text{co-op}}$算法预调度完所有 DAG 未优化任务,并将本次预调度中,第一个被预调度的任务 G_{i-k}作为待优化任务,并且其所选择的资源记为 M_{HEFT};
　/* $\text{MDRS}_{\text{co-op}}$与吞吐量最大化算法 MDRS 不同之处在于:只要出现 DAG 超出其期限约束的情况,则跳出 while do # 循环,$\text{MDRS}_{\text{co-op}}$即终止 */
10:　除该 M_{HEFT}外,将其余所有资源放入待优化任务 G_{i-k}的优化资源集合 $M_{G_{i-k}}$中;
11:　while($M_{G_{i-k}} \neq \Phi$)　do
12:　从 $M_{G_{i-k}}$中顺序取出一个资源 M_q 分配给任务 G_{i-k}后,利用 $\text{MDRS}_{\text{co-op}}$算法预调度所有 DAG 的剩余未优化任务;
13:　if(出现 DAG 完成时间超过其期限约束)then
14:　　DTC(G_{i-k}, M_q) = 0;
15:　else
16:　　DTC(G_{i-k}, M_q) = $\text{Cost}_{\text{Optim}-\text{total}}$ − Cost (G_{i-k}, M_q)$_{\text{total}}$;
17:　end if
18:　　仍将 M_{HEFT}作为任务 G_{i-k}的资源;
19:　end while
20:　if (在 $M_{G_{i-k}}$中存在资源 M_q 使得 DTC(G_{i-k}, M_q) > 0) then
21:　　在 $M_{G_{i-k}}$选择使得 DTC(G_{i-k}, M_q)值最大的 M_q 资源作为 G_{i-k} 的计算资源;
22:　　$\text{Cost}_{\text{Optim}-\text{total}} \leftarrow$ Cost(G_{i-k}, M_q);
23:　　else
24:　　将 M_{HEFT}作为任务 G_{i-k}的资源;
25:　　end if
26:　　将 G_{i-k}从 $G_{i-\text{unoptimized}}$中移至已优化任务集合 $G_{i-\text{optimized}}$中
27:　　if($G_{i-\text{unoptimized}} = \Phi$)　then
28:　　　从 $G_{\text{unoptimized}}$删除对应的 DAG;
29:　　end if
30:　end while
31:　return $G_{i-\text{optimized}}$
32:end Procedure

从以上可以看到,PDTC 优化算法是利用 MDRS 算法中的多个 DAG 之间的相对严格程度来选择要优化的任务,并且在为优化任务试分配其它可选资源时,需用 MDRS 算法来判断是否会出现某 DAG 完成时间超出其期限约束.但需注意的是:由于吞吐量最大化的 MDRS 算法已经对可能出现的"过饱和"情况进行了探测和处理,确定了在时间最小化算法下能在期限内完成的 DAG 及其数量,因此在费用优化阶段所调用的预调度 MDRS 算法与文献[1]中吞吐量最大化的 MDRS 算法有区别:只要出现 DAG 的完成时间超出其期限约束的情况,则跳出 while do # 循环.这个与吞吐量最大化算法 MDRS 有区别的预调度算法,被记为算法 $\text{MDRS}_{\text{co-op}}$,限于篇幅,不再将其列出.

正如文献[1]中对 MDRS 算法时间复杂度的分析,在没有"过饱和"发生情况下,也就是在不发生取消 DAG 的情况下,MDRS 的时间复杂度最大为 $O(n^2 \cdot v^3 \cdot q)$,那么这种情况下的 MDRS 算法与 $\text{MDRS}_{\text{co-op}}$复杂度是一致的.假如在有 n_c 个 DAG 进行调度,每个 DAG 都有 v 个结点任务,机器数量为 q 情况下,PDTC 优化算法最多会调用 $\text{MDRS}_{\text{co-op}}$算法 $n_c \cdot v \cdot q$ 次,因此它们的时间复杂度均为 $O(n_c{}^3 \cdot v^4 \cdot q^2)$,能够在幂次多项式的时间复杂度内提供一个解决方案.

2.4 多 DAG 共享资源调度及费用优化的相关性能指标

针对一组多 DAG 共享资源调度及费用优化问题,为了更好地衡量在不同的算法下多 DAG 完成时间与其各自 Deadline 接近情况和总的执行费用,我们在文献[1]分别提出了相对于 Deadline 的满意度指标 Satisfaction(S)和规范化总费用指标 $N_{C\text{-multiDAG}}(S)$.如果一组 DAG 经过某调度算法 S 调度后,Satisfaction(S)越大,说明经过调度算法 S 的调度后,各 DAG 的完成时间与各自 Deadline 接近程度的平均值更高,同时多个 DAG 之间的这种完成时间与各自 Deadline 接近程度的差异水平也更小,那么该调度算法 S 在相对于 Deadline 的满意度性能指标上的表现也就越好.而规范化总费用指标 $N_{C\text{-multiDAG}}(S)$则衡量的是在某调度算法 S 下,多个 DAG 的总费用与同一组 DAG 所有任务均选择最便宜资源条件下的总费之间的比值,这里不再赘述.以下将通过具

体的调度示例来说明和分析本文所提出的 PDTC 算法的优化方法和文献[1]所提出的 MDRS 算法在上述两方面性能指标上所表现的差异.

3 总费用优化的示例

3.1 示例的有关数据假设

如图 1 所示,假设共有 G_1, G_2, G_3,名称分别为 A、B 和 C 的 3 个不同 DAG 工作流实例需要通过调度器同时在三个处理机器 $M1$、$M2$ 和 $M3$ 资源上调度执行.其中每个 DAG 都有用户指定的最后完成时间的期限约束.假设某 G_i 的期限约束表示为 $t_{\text{Deadline}-G_i}$,且 A、B 和 C 的期限约束分别为 $t_{\text{Deadline}-B}=52$、$t_{\text{Deadline}-B}=73$ 和 $t_{\text{Deadline}-C}=33$.

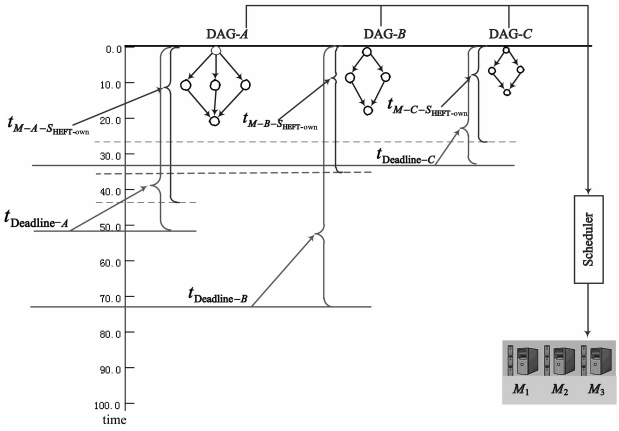


图1 具有期限约束的3个DAG (A、B和C)同时调度示例

以下图 2 中给出了这 3 个 DAG 中任务结点间先后执行的约束关系和数据传递的平均时间 $\overline{c_{i,k}}$;各 DAG 中的任务 n_i在三个处理机器上的执行时间 $w_{i,1}$、$w_{i,2}$、$w_{i,3}$ 和任务 n_i的向上权值 $\text{rank}_u(n_i)$如表 1 所示.

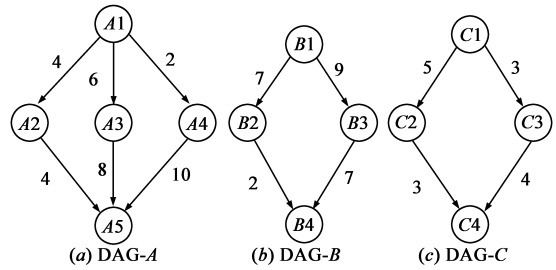


图2 三个DAG的实例

由于 PDTC 的主要调度目标是降低所有 DAG 的总调度执行费用,以下先给出上述实例中 3 个机器的单位时间价格假设.设 $M1$、$M2$ 和 $M3$ 的单位时间价格 p_{M_i}分别为 5、7 和 9.另外,为了比较 PDTC 和 MDRS 算法的规范化费用指标 $N_{C\text{-multiDAG}}(S)$,需要计算某 DAG 的所有任务都选择花费最低资源情况下的总费用 Cost_{C-G_k},即最便宜策略下的费用,因而在表 1 中,根据 $M1$、$M2$ 和 $M3$ 的单位时间价格和任务 n_i 在三个机器上的执行时间 $w_{i,1}$、$w_{i,2}$、$w_{i,3}$,又给出了有关每个任务在三个机器上的最低费用和每个 DAG 的 Cost_{C-G_k}值.

表 1 3 个 DAG 中每个任务在三个机器上的最低费用和每个 DAG 的 Cost_{C-G_k}值

	DAG-A					DAG-B				DAG-C			
/* n_i	A1	A2	A3	A4	A5	B1	B2	B3	B4	C1	C2	C3	C4
$w_{i,1}$	13	11	26	21	7	15	12	13	10	11	13	9	12
$w_{i,2}$	11	10	22	19	6	13	10	12	8	7	11	7	10
$w_{i,3}$	10	9	19	15	5	11	8	11	6	5	9	5	8
$\text{rank}_u(n_i)$	53.7	20.0	36.3	34.3	6.0	49	20.0	27.0	8.0	36	24	21	10
n_i 在 3 个资源上的最低费用	65	55	130	105	35	75	60	65	50	45	65	45	60
Cost_{C-G_k}	390					250				215			

根据以上条件,若利用算法 HEFT,很容易得出 3 个 DAG(A、B、C)分别在这 3 个机器上单独调度的调度长度 M_{own},并分别记为 $t_{M-A-S_{\text{HEFT-own}}}=43$, $t_{M-B-S_{\text{HEFT-own}}}=36$, $t_{M-C-S_{\text{HEFT-own}}}=27$,如图 2 所示.

3.2 PDTC 算法优化过程示例以及与 MDRS 算法调度结果对比与分析

以下表 2 给出了在 MDRS 算法下 3 个 DAG 任务调度过程.表 3 和表 4 描述了上述 3 个 DAG 实例经过吞吐量最大化 MDRS 调度后,继续用 PDTC 对 MDRS 调度结果进行费用优化过程、相关调度结果及性能指标.图 4 则给出了 PDTC 与 MDRS 两算法的调度图对比.

表 3 中,符号"√"表示算法对每个任务所选中的资源.在 PDTC 算法栏中,第一轮用 $\text{MDRS}_{\text{co-op}}$预调度后同样选择 $A1$ 任务作为第一个要优化的任务,$A1$ 选择的 M_{HEFT}资源为 $M3$,那么除了 $M3$ 外,$A1$ 的优化资源集合 $M_{G_{i-k}}$中的资源 M_q 有 $M1$ 和 $M2$ 两个.如果将 $M1$ 分配给 $A1$,利用 $\text{MDRS}_{\text{co-op}}$调度其余所有待优化任务,则出现了 A 超出其期限约束的情况,此时,与当前最低总费用相比的总费用降低量 $\text{DTC}(G_{i-k}, M_q)$为 $\text{DTC}(A1, M1)$,并将其置为 0.如果将 $A1$ 分配给 $M2$,利用 $\text{MDRS}_{\text{co-op}}$预调度其余所有未优化任务后,所有 DAG 均能在期限内完成,并且 $\text{Cost}(A1, M2)_{\text{total}}=1009$,那么总

费用降低量 DTC($A1$, $M2$) = $\mathrm{Cost}_{\mathrm{Optim-total}}$ − Cost($A1$, $M2$)$_{\mathrm{total}}$ = 27(初始当前最低总费用 $\mathrm{Cost}_{\mathrm{Optim-total}}$ = $\mathrm{Cost}_{\mathrm{M\text{-}DRS-total}}$ = 1036),因此根据算法,选择 $M2$ 作为 $A1$ 优化后的资源,并且当前最低总费用 $\mathrm{Cost}_{\mathrm{Optim-total}}$被置为 1009. 重复上述过程,直到所有任务优化完毕.

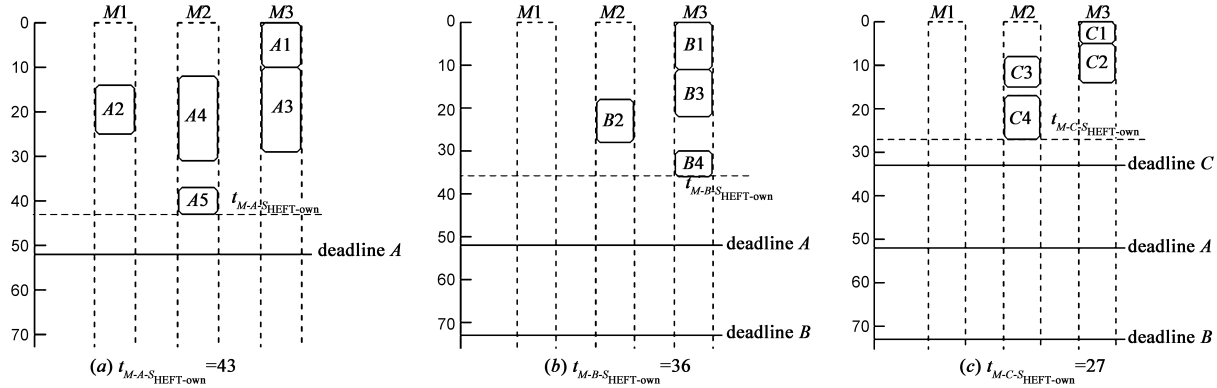


图3　利用HEFT分别单独调度每个DAG的调度图及相应DAG的M_{own}

表 2　MDRS 对 3 个 DAG 的调度过程

	根据 DAG 剩余任务相对严格程度所选择的任务	三个 DAG 调度过程中的剩余任务相对严格程度(见文献[1])		
		$r_{un(A)-S_{\mathrm{HEFT}}}$	$r_{un(B)-S_{\mathrm{HEFT}}}$	$r_{un(C)-S_{\mathrm{HEFT}}}$
任务调度选择依据	$A1$	0.8269	0.4931	0.8182
	$C1$	0.7857	0.5205	0.8788
	$C2$	0.7857	0.5616	0.8462
	$C3$	0.8095	0.5616	0.8261
	$A3$	0.9000	0.5616	0.7143
	$A4$	0.9000	0.5616	0.7143
	$A2$	0.8824	0.6909	0.7143
	Pop out($A2$)	0.5556	0.8444	2.000
	—	用 HEFT 调度完 C 中所有剩余未调度任务(此时 C 中只剩有 $C4$ 任务)		
	$B1$	0.8261	0.8636	C 已调度完毕
	$B3$	0.7778	0.8065	—
	$A2$	0.7778	0.7500	—
	$B2$	0.5556	0.7500	—
	$B4$	0.5556	0.5714	—
	$A5$	0.5556	B 已调度完毕	—
任务调度顺序	$A1$, $C1$, $C2$, $C3$, $A3$, $A4$, $A2$, Pop out($A2$), $C4$, $B1$, $B3$, $A2$, $B2$, $B4$, $A5$			
每个 DAG 的完成时间 A	48			
每个 DAG 的完成时间 B	67			
每个 DAG 的完成时间 C	29			
Satisfaction(S)	45.7879			

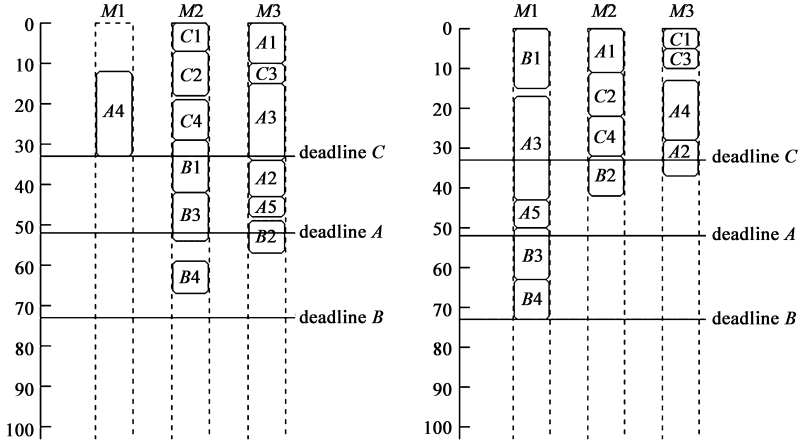
表 3　PDTC 对 3 个 DAG 的费用优化过程

G_{i-k}	M_{HEFT}	$M_{G_{i-k}}$	M_q	DTC(G_{i-k}, M_q)	$\mathrm{Cost}_{\mathrm{Optim-total}}$
$A1$	$M3$	$M1$; $M2$	$M1$	0(A 超出期限约束)	1036 ($\mathrm{Cost}_{\mathrm{MDRS-total}}$)
			$M2$ √	27	1009
$C1$	$M3$ √	$M1$; $M2$	$M1$	0(C 超出期限约束)	1009
			$M2$	0(C 超出期限约束)	
$C2$	$M3$	$M1$; $M2$	$M1$	0(C 超出期限约束)	983
			$M2$ √	26	
$C3$	$M3$ √	$M1$; $M2$	$M1$	0(C 超出期限约束)	983
			$M2$	0(C 超出期限约束)	
$A3$	$M3$	$M1$; $M2$	$M1$ √	1	982
			$M2$	-37	
$A4$	$M3$ √	$M1$; $M2$	$M1$	0(A 超出期限约束)	982
			$M2$	0(A 超出期限约束)	
$C4$	$M2$ √	$M1$; $M3$	$M1$	0(C 超出期限约束)	982
			$M3$	0(C 超出期限约束)	
$A2$	$M3$ √	$M1$; $M2$	$M1$	0(A 超出期限约束)	982
			$M2$	0(A 超出期限约束)	
$A5$	$M1$ √	$M2$; $M3$	$M2$	0(A 超出期限约束)	982
			$M3$	0(A 超出期限约束)	
$B1$	$M1$ √	$M2$; $M3$	$M2$	-16	982
			$M3$	-35	
$B3$	$M2$	$M1$; $M3$	$M1$ √	27	955
			$M3$	−11	
$B2$	$M2$ √	$M1$; $M3$	$M1$	0(B 超出期限约束)	955
			$M3$	−2	
$B4$	$M1$ √	$M2$; $M3$	$M2$	0(B 超出期限约束)	955
			$M3$	0(B 超出期限约束)	

从表 4 中的优化结果来看,3 个 DAG 在吞吐量最大化的 MDRS 算法下的总费用为 1036,而在 PDTC 算法下,A、B 和 C 三个 DAG 的总费用为 955.从这三个算法的规范化总费用 $N_{C\text{-multiDAG}}(S)$来看,根据 Cost_{C-G_k}的含义,结合表 1 的 3 个 DAG 在不考虑期限约束的最便宜策略下的费用 Cost_{C-G_k}(分别为 390、250 和 215),则这 3 个 DAG 的 $\sum_{k=1}^{n_c}\mathrm{Cost}_{C-G_k}=855$,根据表中 MDRS 和 PDTC 两个算法下的总费用 $\sum_{k=1}^{n_c}\mathrm{Cost}_{S-G_k}$,可得到 MDRS 的规范化费用约为 1.2117,而在 PDTC 算法下约为 1.1170.从表 1 中资源的价格和任务执行时间参数中可以看到,$M1$ 价格最低,执行任务的时间也最长,$M3$ 价格最高,执行任务所需要的时间也最短,绝大部分任务的花费也基本是在价格较高的资源上的花费也较高.另外,在表 4 中,MDRS 算法下的 Satisfaction(MDRS) = 45.7879,而经过 PDTC 算法费用优化后,这 3 个 DAG 的完成时间比 MDRS 算法下的完成时间更加接近各自的期限,该指标提升至 59.0499,进一步说明了费用优化算法 PDTC 较为充分地利用了各 DAG 的冗余时间,选择了更多执行时间较长而花费较低的资源.

表 4 MDRS 和 PDTC 两算法调度结果对比

调度结果指标	MDRS	PDTC
A 的完成时间	48	50
B 的完成时间	67	73
C 的完成时间	29	32
不同算法的总费用 $\sum_{k=1}^{n_c}\mathrm{Cost}_{S-G_k}$	1036	955
各算法的规范化总费用 $N_{C\text{-multiDAG}}(S)$ $(\sum_{k=1}^{n_c}\mathrm{Cost}_{C-G_k}=855)$	1.2117	1.1170
M_1 的资源利用率 U_{M1}	28.76%	97.26%
M_2 的资源利用率 U_{M2}	83.56%	57.53%
M_3 的资源利用率 U_{M3}	76.71%	46.58%
Satisfaction(S)	45.8931	59.0499



(*a*) MDRS算法调度图 (*b*) PDTC算法调度图

图4 3个DAG实例在MDRS和PDTC两算法下的调度图

从图 4(*a*)MDRS 算法调度图与图 4(*b*)PDTC 算法调度图中所有任务在各个资源上分布情况的对比来看,MDRS 算法属于时间最小化算法,在速度较快的 $M3$ 和 $M2$ 上分布的任务较多,而经 PDTC 算法优化后,分布情况发生了变化,即在速度越快、价格越高的机器资源上分布的任务数量也越少,资源利用率也越低.在图 4(*b*)中,速度较慢价格也较低的 $M1$ 上分布的任务最多,资源被占用的时间也最长,这也直观地反映了 PDTC 费用优化算法进行了有效的费用优化.

4 实验与分析

本文所提出和解决的问题是在 DAG 吞吐量最大化的基础上进行费用优化,而 PDTC 费用算法采取的方案正是在 MDRS 算法对一组具有期限约束的多 DAG 进行吞吐量最大化调度基础上进行的费用优化,算法 PDTC 与 MDRS 在 DAG 完成率指标上完全一致.鉴于此,本章实验主要是在规范化总费用指标 $N_{C\text{-multiDAG}}(S)$和相对于期限的满意度指标 Satisfaction(S)两方面性能指标上对 PDTC 和 MDRS 两算法的性能进行验证对比.

4.1 实验设置

本文沿用了文献[1]有关 MDRS 算法的实验对象和相关结果,其中包括 100 组随机生成的 DAG 及其对应的随机选择的机器资源数量、在机器上的每组 DAG 中每个任务的执行时间、每组 DAG 对应的 7 个不同的期限约束组.除了这些实验对象及相关数据外,本文实验还需要

对每组 DAG 调度目标资源的单位时间价格进行设定.在文献[1]的实验中,每组 DAG 的任务在某数量机器上的运行时间由异构因子参数集合 SET_β 中的一个参数生成,并按升序排序后依次作为任务在各机器上的运行时间.然而,DAG 内的任务在不同资源上的执行时间差异越大,表明各机器间的速度性能差异越大,那么资源之间的价格差异也应该越大,并且执行时间越长的机器,表明机器的速度及性能越低,价格也应该越低.为了体现这一计价模式的特点,当某个 β 参数被选择为某组 DAG 任务在对应机器组上的预计执行时间的生成参数,那么这组机器的单位价格差异程度也应该与这组 DAG 所选择的 β 参数一致.本文的实验中,针对文献[1]的实验对象中每组 DAG 所对应资源组的价格生成,采取了以 7 为资源价格的均值和以该组 DAG 所选的 β 参数为资源价格方差的正态分布 $N(7,\beta)$随机生成这组资源的单位时间价格数据,然后将这组数据进行降序排序依次作为该组资源的价格(因为任务在这组资源上的运行时间为升序排序,即需要保证如表 1 所示:任务在机器上的执行时间与机器的价格变化趋势相反).

4.2　实验结果及分析

以下为 MDRS 和费用优化算法 PDTC 两算法对 100 组 DAG 分别在 7 种不同类型期限约束条件下调度的规范化总费用 $N_{C\text{-multiDAG}}(S)$均值和 Satisfaction(S)均值的情况,分别如图 5 和图 6 所示.

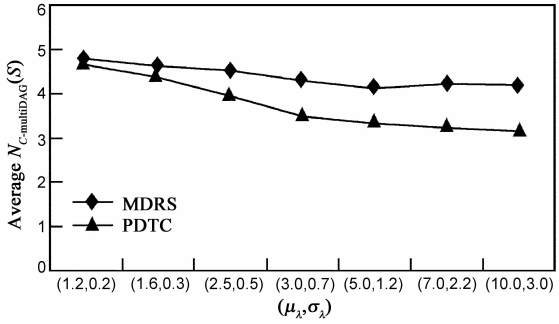


图5　7种$(\mu_\lambda,\sigma_\lambda)$参数下的2种算法的规范化总费用 $N_{C\text{-multiDAG}}(S)$均值

图 5 为 MDRS 和 PDTC 两算法的 100 组 DAG 分别在 7 种期限约束参数下的规范化总费用 $N_{C\text{-multiDAG}}(S)$均值.当 DAG 期限的紧急程度较高和差异水平较小时(也即 μ_λ 和 σ_λ 两个参数取值都较小时),2 种算法之间的规范化总费用 $N_{C\text{-multiDAG}}(S)$均值没有明显差异,且都超过了 4.5.这表明:为了能保证各 DAG 都能在期限约束内完成,两算法都尽可能选择最早完成时间的资源.对 PDTC 来说,选择任务的 M_{HEFT}以外的资源容易造成 DAG 的完成时间超出其期限约束,费用优化的空间较小.从两算法该指标的变化趋势来看,随着$(\mu_\lambda,\sigma_\lambda)$中 μ_λ 和 σ_λ 取值的增大,MDRS 算法下的规范化总费用 $N_{C\text{-multiDAG}}(S)$的均值没有明显的变化趋势,这就证明了由于 MDRS 算法的调度目标为 DAG 的吞吐量最大化,虽然期限紧急程度的降低和差异水平的扩大会影响各个 DAG 的期限紧急程度关系的变化,使得 DAG 的任务调度顺序发生变化,DAG 的完成率和总费用也会上升,但所增加的冗余时间并不能有效地被利用,因此规范化总费用 $N_{C\text{-multiDAG}}(S)$并没有发生明显变化.而 PDTC 算法在每种情况下的规范化费用 $N_{C\text{-multiDAG}}(S)$均值都比 MDRS 低,说明该算法能够有效地对 MDRS 调度后的结果进行有效的总费用优化.并且随着 μ_λ 和 σ_λ 取值的增大,PDTC 的 $N_{C\text{-multiDAG}}(S)$指标均值总体上都处于下降趋势,说明期限紧急程度越低,给予 DAG 的冗余时间越多,该算法调度后的总费用也会越低,表明该算法能较好地根据各 DAG 期限的紧急程度,在保证各 DAG 在期限内完成的约束条件下,利用冗余时间而选择了花费较低的资源.但随着期限紧急程度低到一定程度的时候,如(3.0,0.7)以后,PDTC 的规范化费用指标也基本停止了降低.这也证明了进一步延长一组 DAG 的完成期限而增加它们的冗余时间,对费用降低的贡献将会越来越小,因此 PDTC 适用于各 DAG 期限紧急程度较高情况下的总费用优化.

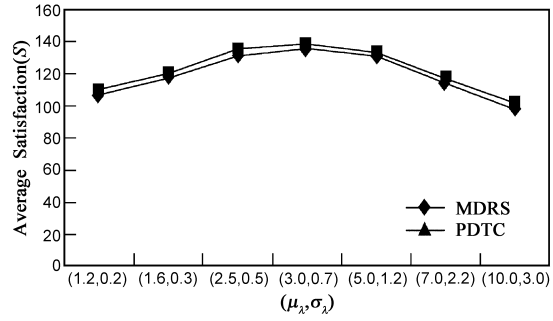


图6　7种$(\mu_\lambda,\sigma_\lambda)$参数下的2种算法的Satisfaction($S$)均值

图 6 显示了在不同的期限紧急程度和差异水平下的 MDRS 和 PDTC 两种算法的平均 Satisfaction(S)值.可以看出,随着$(\mu_\lambda,\sigma_\lambda)$中参数的取值不同,这两种算法下的满意度指标的变化趋势基本一致,但 PDTC 的均值要高于 MDRS.这表明:与 MDRS 算法相比,PDTC 更能充分地利用各 DAG 的冗余时间,更多的任务选择了执行时间较长而花费较低的资源,致使各个 DAG 的完成时间进一步接近了各自的期限.

从上述实验结果分析来看,本章提出的 PDTC 算法对一组 DAG 利用总费用变化量探测的方法进行费用优化,不仅能够保证 DAG 组中的每个 DAG 都能在期限之前完成,又能利用每个 DAG 的期限所留下的冗余时间降低所有 DAG 总费用优化.

5 小结

本文首先论述了多 DAG 混合调度费用优化与现有有关单 DAG 费用优化的区别及难点,然后在文献[1]关于多 DAG 混合调度吞吐量最大化 MDRS 算法的基础上,对多 DAG 的调度进行费用优化问题提出了基于总费用变化量探测的 PDTC 费用优化算法.实验证明,本章所提出的方法能够有效降低吞吐量最大化后的多个 DAG 的费用.

参考文献

[1] 田国忠.多 DAG 共享资源调度的若干问题研究[D].北京:北京工业大学,2014.
Tian Guozhong. Research on Several Problems of Scheduling Multiple DAGs Sharing Resources[D]. Beijing: Beijing University of Technology, 2014. (in Chinese)

[2] 苑迎春,李小平,王茜,王克俭.成本约束的网格工作流时间优化方法[J].计算机研究与发展,2009,46(2):194 – 201.
Yuan Yingchun, Li Xiaoping, Wang qian, and Wang Kejian. Time Optimization Heuristics for Scheduling Budget-Constrained Grid Workflows[J]. Journal of Computer Research and Development, 2009, 46(2): 194 – 201. (in Chinese)

[3] Abrishami S, Naghibzadeh M, Epema D. Cost-driven scheduling of grid workflows using partial critical paths[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 23(8): 1400 – 1414.

[4] Abrishami S, Naghibzadeh M, Epema D. Deadline-constrained workflow scheduling algorithms for IaaS Clouds[J]. Future Generation Computer Systems, 2013, 29(1): 158-169.

[5] Ming Mao, Humphrey M. Auto-scaling to minimize cost and meet application deadlines in cloud workflows[A]. Proc of the International Conference High Performance Computing, Networking, Storage and Analysis (SC)[C]. Seattle, Washington: IEEE Computer Society, 2011. 1 – 12.

[6] Luiz F Bittencourt, Edmundo R M Madeira. HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds[J]. Journal of Internet Services and Applications, 2011, 2(3): 207 – 227.

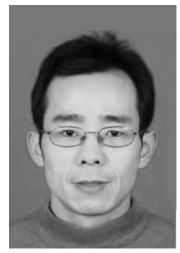
[7] Georgios L, Stavrinides Helen D. Scheduling multiple task graphs with end-to-end deadlines in distributed real-time systems utilizing imprecise computations[J]. Journal of Systems and Software, 2010, 83(6): 1004 – 1014.

[8] Georgios L, Stavrinides Helen D. Scheduling real-time DAGs in heterogeneous clusters by combining imprecise computations and bin packing techniques for the exploitation of schedule holes[J]. Future Generation Computer Systems, 2012, 28(7): 977 – 988.

[9] TOPCUOGLU H, Hariri S, Min-You W. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. Parallel and Distributed Systems, IEEE Transactions on, 2002, 13(3): 260 – 274.

[10] 陈宏伟,王汝传.费用-时间优化的网格有向无环图调度算法[J].电子学报,2005,33(8):1375 – 1380.
Chen Hongwei, Wang Ruchuan. A grid DAG scheduling algorithm for cost-time optimization[J]. Acta Electronica Sinica, 2005, 33(8): 1375 – 1380. (in Chinese)

作者简介



田国忠 男,1971 年出生,博士,副教授,主要研究领域为并行与分布计算,高性能集群计算.
E-mail: tiangz_bjut@qq.com

肖创柏 男,1962 年出生,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络安全,信号处理,模式识别,云计算.
E-mail: cbxiao@bjut.edu.cn

谢军奇 男,1990 年出生,硕士生,主要研究领域为高性能集群计算.