计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140205
52(6):1400-1408,2015

带通信开销的 DAG 工作流费用优化模型与算法

郭 禾[1] 陈 征[1] 于玉龙[1] 王宇新[2] 陈 鑫[1]
[1](大连理工大学软件学院 辽宁大连 116620)
[2](大连理工大学计算机科学与技术学院 辽宁大连 116024)
(guohe@dlut.edu.cn)

A Communication Aware DAG Workflow Cost Optimization Model and Algorithm

Guo He[1], Chen Zheng[1], Yu Yulong[1], Wang Yuxin[2], and Chen Xin[1]
[1](*School of Software*, *Dalian University of Technology*, *Dalian*, *Liaoning* 116620)
[2](*School of Computer Science and Technology*, *Dalian University of Technology*, *Dalian*, *Liaoning* 116024)

Abstract Communication overhead can not be neglected in cloud environment. However, without considering communication overhead among tasks, a cost optimization model of DAG(directed acyclic graph) workflow is difficult to apply in the actually cloud environment. Therefore, this paper puts forward a cost optimization model of DAG workflow with communication overhead. In addition, based on the hierarchical algorithm, which distributes the tasks into groups based on levels and schedules them by level, the paper proposes a cost optimization awared communication algorithm (CACO). CACO uses the forward consistent (FC) rules to solve the minimum completion time of the workflow. Also, by using the bottom hierarchical strategy to divide the task into separated layers, CACO transfers the cost optimization problem from the whole to the part. Furthermore, in order to increase the space of cost optimization and improve the results, CACO adopts dynamic programming method to collect discrete "time pieces" that is produced during the selecting services. The simulation results show that, compared with DTL(deadline top level), DBL(deadline bottom level), TCDBL(temporal consistency deadline bottom level), CACO has greatly enhanced the cost optimization effect considering communication overhead.

Key words communication overhead; cost optimization; workflow; hierarchical; DAG schedule

摘 要 通信开销在云环境中无法忽略,但现有 DAG(directed acyclic graph)工作流费用优化模型大都未考虑任务之间的通信开销,难以在实际云环境中应用.为此,提出带通信开销的工作流费用优化模型 CA-DAG(communication aware-DAG),并在分层算法的基础上提出针对 CA-DAG 模型的调度算法 CACO(communication aware cost optimization).CACO 使用前向一致规则(forward consistent, FC)求解工作流的最小完工时间;根据逆向分层策略将任务分层,使费用优化问题从全局转化到局部;采用动态规划方法收集任务在选择服务时产生的零散"时间碎片",增加任务的费用优化空间,改善费用优化效果.仿真实验结果表明,在考虑通信开销时,CACO 费用优化效果较 DTL(deadline top level),DBL(deadline bottom level),TCDBL(temporal consistency deadline bottom level)都有显著提高.

关键词 通信开销;费用优化;工作流;分层;DAG 调度

中图法分类号 TP393

收稿日期:2014-03-10;**修回日期**:2014-12-08
基金项目:国家自然科学基金项目(61300016,11372067)
通信作者:陈 鑫(cx.dlut@gmail.com)

在云环境中对服务进行付费已经成为一种趋势.云服务提供商在大型服务器上部署多种服务[1],根据服务的 QoS(quality of service)属性制定收费标准,用户使用这些服务来完成提交的应用.这些应用多以工作流的形式存在,包含若干个任务,每个任务通常对应多个服务以供选择.云环境对工作流调度时应综合考虑执行时间、执行费用、可靠性等多个因素.通常云服务提供商必须在用户给定的截止期限内完成其提交的工作流,并对工作流中的任务进行合理的调度来优化执行的总费用,从而提高用户对服务质量的满意度.因此研究在云环境中基于截止期约束的工作流费用优化问题很有必要.

基于截止期约束的工作流费用优化问题,已有多个学者进行了大量的研究,多基于分层算法.苑迎春等人[2-3]提出的截止期约束的逆向分层费用优化算法(deadline bottom level, DBL)基于逆向分层,将截止期转化为层截止时间,逐层进行优化;在此基础上又提出时间约束的前向串归的算法(forward serial reduction with deadline, FSRD),将可串规约组放入一层,提高分层算法的优化效果;Yuan 等人[4]提出截止期约束的最早树算法,使用串规约方法只为关键任务选择服务,然后对非关键任务的约束时间进行设置.刘灿灿等人[5-6]为减少调度过程中产生的"时间碎片",增大费用优化空间,在分层算法的基础上,提出基于时序一致的截止期约束逆向分层算法(temporal consistency deadline bottom level, TCDBL)和基于路径平衡的费用优化算法(path balance based cost optimization, PBCO),TCDBL 研究工作流的时序特征,基于任务的一致性状态对费用进行优化;PBCO 对工作流中各路径长度进行调整,基于路径平衡进行费用优化.除了分层算法,苑迎春等人[7]提出基于优先级的调度算法以及刘灿灿等人[8]对其进行的改进,提出基于时间藕合强度(best fit with time-dependent coupling strength, BFTCS)和时间藕合强度及时间灵活度(best fit based on time-dependent coupling strength and temporal mobility, BFTCSTM)规则的迭代算法,以一定的规则迭代地计算每个任务的优先级来确定优化顺序,以此求得最优调度方案.此外,还有基于遗传算法、粒子群优化算法、蚁群算法等[9-11]的启发式费用优化算法.这些算法虽然取得了良好的费用优化效果,但都没有考虑到任务与任务之间的通信开销.

云计算通常将服务资源部署在大型集群上,用户提交的工作流中的任务会使用多个节点上的资源,任务之间有着频繁的数据交互.然而节点间的网络传输带宽有限,任务间跨节点的数据传输意味着不可忽视的通信开销[12],应将其加入到模型中.一些学者研究有向无环图(directed acyclic graph, DAG)工作流的调度时考虑到了通信开销,N'takpé 等人[13]、Hönig 等人[14]、Zhao 等人[15]和 Stanzani 等人[16]提出的调度算法都基于 HEFT[17](heterogeneous earliest-finish-time)并对此算法改进,但其目标均为单一优化最小完成时间,没有考虑到云环境中各任务的执行费用问题;Choudhary 等人[18]和 Gao 等人[19]针对每个用户提交的不同 QoS 要求量身定制调度方案,此方法需要用户和云服务提供商遵守服务等级协议(service level agreement, SLA)约束,算法复杂性较高;Kumar 等人[20]和 Gao 等人[21]提出的调度算法都考虑到了时间和费用,前者不考虑用户截止期约束,以期望得到时间和费用之间的均衡解,后者则更多地关注多用户之间的资源分配.

综上,本文将不可忽视的通信开销加入到工作流费用优化模型中,提出带通信开销的工作流(communication aware-directed acyclic graph, CA-DAG)费用优化模型;并提出针对该模型的费用优化调度算法 CACO(communication aware cost optimization).CACO 基于分层算法[22],采用前向一致规则(forward consistent, FC)求解最小完工时间,得到准确的截止时间下限,作为有无解的判断;在任务调度过程中,使用动态规划的方法[23]迭代地调整任务的最终最早开始时间和最晚结束时间,收集任务在选择服务时产生的零散"时间碎片",增加任务的执行时间窗口,提高费用优化效果.实验表明,在考虑通信开销时,算法 CACO 的费用优化效果较截止期约束的正向分层算法(deadline top level, DTL),DBL,TCDBL 分别平均提高了 59%,47%,29%.

1　带通信开销的工作流费用优化模型

带通信开销的工作流费用优化模型基于 DAG,$G=\langle V,E,S\rangle$可形象表示为图 1 和表 1,其中 $V=\{0,1,2,\cdots,n\}$为节点的集合,表示所有的任务;$E=\{e_{ij}\}(i,j\in V)$为有向边的集合,表示任务之间的控制或数据依赖,同时称 i 为 j 的前驱,j 为 i 的后继.i 的所有前驱形成前驱集合,记为 $pre(i)\subseteq V$;所有后继形成后继集合,记为 $succ(i)\subseteq V$.$|e_{ij}|$为任务 i,j 之间数据传输所需时间.为了简化算法设计,只考虑

单入单出的 DAG,对于非单入单出的 DAG,通过增加虚任务,如图 1 中的节点 0 和 7(执行时间和费用都为 0 的任务,同时与其他任务之间的通信开销为 0),来转换成单入单出的 DAG.

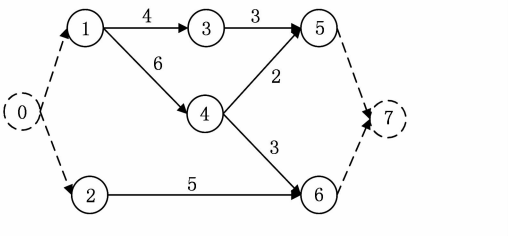


Fig. 1 The instance of workflow(DAG).

图 1 工作流应用实例(DAG 图表示)

Table1 The Services Set of the Activities in Fig. 1

表 1 图 1 中各活动的候选服务集合

Task	$S(i)$	$\lvert S(i)\rvert$
1	{⟨3,9,2⟩,⟨4,6,3⟩,⟨8,2,1⟩}	3
2	{⟨4,8,3⟩,⟨7,5,1⟩}	2
3	{⟨2,9,1⟩,⟨4,7,2⟩,⟨6,5,4⟩,⟨9,2,3⟩}	4
4	{⟨2,7,1⟩,⟨3,5,4⟩,⟨5,3,2⟩}	3
5	{⟨3,8,4⟩,⟨4,7,1⟩,⟨9,2,2⟩}	3
6	{⟨4,7,3⟩,⟨6,5,1⟩,⟨7,3,4⟩}	3

$G=\langle V,E,S\rangle$中的 S 为所有的服务集合,如表 1. 每个任务 i 都对应 1 个候选服务集合 $S(i)=\{S_{ik}\mid S_{ik}\in S\}$,$k\in[1,|S(i)|]$;服务集合的规模为$|S(i)|$,表示该任务共有多少个可选服务. $S_{ik}=\langle t,c,m\rangle$表示可以完成任务 i 的 1 个服务,其中 $S_{ik}.t$ 为其执行时间(使用此服务执行任务 i 所需时间);$S_{ik}.c$ 为其执行费用(使用此服务执行任务 i 所需费用);由于 CA-DAG 带有通信开销,引入 $S_{ik}.m$ 表示服务所在的物理机标识.

通信开销为任务之间进行数据传输时所需的时间. 当 2 个具有数据依赖的任务所选的服务位于不同的物理机时,就会产生通信开销. $\forall i,j\in V$ 且 $i\in pre(j)$,若任务 i,j 分别选定服务 $S_{ip}\in S(i)$,$S_{jq}\in S(j)$,则此时 2 个任务之间的通信开销如式(1)所示:

$$Trans(i,j)=\begin{cases}0, & S_{ip}.m=S_{jq}.m;\\ |e_{ij}|, & \text{otherwise.}\end{cases}\tag{1}$$

CA-DAG 模型表示的工作流的调度即为各任务在对应的候选服务集合中选择 1 个服务. 各任务的服务选定后,任务间通信时间随之确定,任务 0 到任务 n 之间存在 1 条最长路径(称为关键路径),其路径长度(路径长度为该路径上所有任务的执行时间和通信时间之和)为工作流的执行时间;各任务所选服务的费用之和为工作流的执行费用. 本文的调度目标即为 V 中的任意任务 i 在对应的服务集合 $S(i)$中选择 1 个服务 S_{ik},使得整个工作流的执行时间在约束时间(记为 δ)之内,执行费用最小,形式化描述如式(2)~(5):

$$\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n),\tag{2}$$

$$\text{s.t. } \boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{i|S(i)|})^{\mathrm{T}},\text{且 }|\boldsymbol{x}_i|=1;\tag{3}$$

$$\delta_j\geqslant\delta_i+Trans(i,j)+\sum_{k=1}^{|S(j)|}|S_{jk}.t\times\boldsymbol{x}_j|,\ j\in succ(i);\tag{4}$$

$$\min\sum_{i=1}^{n}\sum_{k=1}^{|S(i)|}|S_{ik}.c\times\boldsymbol{x}_i|,\ \delta_n\leqslant\delta.\tag{5}$$

其中,$\boldsymbol{x}_i$ 为选择向量,表示每个任务只能且必须选择 1 个服务;式(4)表示任务之间的偏序关系;式(5)表示调度目标,在约束时间内使得工作流的执行费用最小;$\boldsymbol{X}$ 为最终的调度方案,使用选择向量代表各任务选择的服务.

2 前向一致规则

调度工作流进行费用优化需要确定整个工作流的最小完工时间(使得关键路径最短的工作流的执行时间),用户给定截止期必须大于或等于该值才能对任务调度. 在现有模型中通常采用最小关键路径规则(minimum critical path, MCP)[3]求解工作流的最小完工时间,MCP 将关键路径上任务的最小执行时间累加作为工作流的最小完工时间. 然而加入通信开销后,在任务调度之前无法确定执行任务的物理机,通信开销为不定值. 例如图 1 中,若任务 1,3 选择同一物理机上的服务,则通信开销为 0;否则通信开销为 4. 此时采用 MCP 求解最小完工时间时,若只考虑到关键路径上任务执行时间而忽略掉通信开销,则求出的值会小于真正的最小完工时间;若将关键路径上的通信开销简单地加入,又会大于真正的最小完工时间,因此,MCP 不能用于 CA-DAG 模型.

提出 FC 求解 CA-DAG 模型的最小完工时间. FC 采用递归的方法确定每个任务的最快服务(可以使任务最早完成的服务)及最早完成时间,只有当此任务的所有前驱任务的最快服务都确定下来,才能确定此任务的最快服务.

按照 FC 求解工作流的最小完工时间需要确定工作流中各任务的最早完成时间及此时任务所选服

务,当该任务的前驱集合为空集时选择执行时间最短的服务,执行此服务的完成时间即为最早完成时间;否则,顺序遍历该任务服务集合中的每一服务,计算选择该服务时与所有前驱任务的通信时间,得到此任务执行该服务的完成时间,使得完成时间最早的服务为该任务的最快服务,执行此服务得到的完成时间为该任务的最早完成时间.综上可以得出,$\forall i\in V$,任务 i 的最早完工时间 $T_{ef}(i)$ 及最快服务 S_i^* 可如式(6)(7)计算.工作流的最小完工时间为最后 1 个任务的最早完成时间,记为 $T_{min}=T_{ef}(n)$.

$$\begin{cases} S_i^* = \{S_{ik} \mid \min\limits_{0<k\leqslant|S(i)|} S_{ik}.t\}, \\ T_{ef}(i) = S_i^*.t, \end{cases} \quad pre(i)=\varnothing. \tag{6}$$

$$\begin{cases} S_i^* = \{S_{ik} \mid \min\limits_{0<k\leqslant|S(i)|} \{\max\limits_{j\in pre(i)}\{T_{ef}(j) + \\ \quad S_{ik}.t + Trans(j,i)\}\}\}, \\ T_{ef}(i) = \max\limits_{j\in pre(i)}\{T_{ef}(j) + S_i^*.t + \\ \quad Trans(j,i)\}, \end{cases} \quad \text{otherwise.} \tag{7}$$

3　带通信开销的费用优化算法

CACO 基于分层算法,在引入通信开销后,由于任务之间通信开销的不确定,加入动态分层.CACO 调度 CA-DAG 模型表示的工作流进行费用优化分为 3 步:1)使用 FC 确定工作流的最小完工时间,作为判断此工作流是否可以调度的依据;2)分层阶段,采用已有的逆向分层算法,将固定的通信开销(即不考虑 2 个任务是否在同 1 台物理机上执行)加入到分层算法中,并得到分层最小完工时间作为调度阶段的初始状态;3)调度阶段,使用迭代方式重新调整已调度任务的最终完成时间以及未调度任务的最终最早开始时间,确定未调度任务的时间窗口,进一步收集零散“时间碎片”.

3.1　分层阶段

分层是将用户给定截止期约束 δ 分解为任务的时间窗口 $W1:[\beta_i,\delta_i]$,将全局费用优化问题转换成任务的局部费用优化问题.时间窗口指任务的最早开始时间和最晚结束时间组成的 1 个时间范围,任务选择的服务执行时间需小于此时间范围.CACO 基于 DBL[3] 中提出的逆向分层,给定图 $G=\langle V,E,S\rangle$,任务 i 的逆向深度 $BD(i)$ 定义为 i 到出口任务 n 的最长路径长度上有向边的个数;将相同深度值的任务划分到同层,第 m 层包含的任务为 $BL(m)=\{i\mid BD(i)=m,i\in V\}(0\leqslant m\leqslant BD(1))$.

分层结束后,计算各任务所允许的最早开始时间 β_i、最晚结束时间 δ_i 和各层的层截止时间 $\delta_{BL(m)}$,位于同层的任务具有相同的最晚结束时间,都为该层的层截止时间,因此,计算如式(8)所示.其中,$\delta_{BL(m)}$ 为第 m 层的层截止时间;T_{BLmin} 为最后一层的层截止时间,称为分层最小完工时间.当用户给定截止期 $\delta\geqslant T_{BLmin}$ 时,才能对任务进行分层;当 $\delta>T_{BLmin}$ 时,存在冗余时间,此时将冗余时间平均分配到各层,各层得到的冗余时间为 T_{LS},T_{BLmin} 是在 $T_{LS}=0$ 时按照式(8)计算得出的.

$$\begin{cases} \beta_i = \begin{cases} 0, & pre(i)=\varnothing, \\ \max\limits_{j\in pre(i)}\{\delta_j + |e_{ji}|\}, & \text{otherwise}, \end{cases} \\ \delta_i = \delta_{BL(m)} = \max\limits_{i\in BL(m)}\{\beta_i + \min\limits_{0<k\leqslant|S(i)|}\{S_{ik}.t\}\} + T_{LS}. \end{cases} \tag{8}$$

由于冗余时间 T_{LS} 在分层算法初始时无法立即确定,只有在假设 $T_{LS}=0$ 时确定 T_{BLmin} 后才能求得 T_{LS}.因此,分层算法首先求出不带 T_{LS} 的各任务的时间窗口 $W1:[\beta_i,\delta_i]$,得到分层最小完工时间 T_{BLmin},当 $\delta>T_{BLmin}$ 时,再求得最终的时间窗口 $W1:[\beta_i,\delta_i]$,算法的伪码描述如下:

算法 1. CACO 分层阶段.

① 令 $T_{LS}=0$ 带入式(8),求得各任务的时间窗口 $W1:[\beta_i,\delta_i]$;
② $T_{BLmin}\leftarrow\delta_n$;
③ $T_{LS}\leftarrow(\delta-T_{BLmin})/BL(0)-2$;
④ if ($T_{LS}>0$)
⑤　将 T_{LS} 带入式(8),重新求得各任务的时间窗口 $W1:[\beta_i,\delta_i]$;
⑥ endif

3.2　调度阶段

对各任务进行调度时,采用动态规划的方法迭代地调整各任务的时间窗口.动态规划就是将过程分成若干个互相联系的阶段,在它的每个阶段都需要作出决策,从而使整个过程达到最好的效果.在本文中,每个任务在选择服务时都受到其前驱任务的影响,当其所有的前驱任务调度完成后,根据各前驱任务的最终完成时间重新确定该任务的最早开始时间.该任务调度完成后,根据所选服务的实际所需时间,修改其最终完成时间,这样就可以充分利用各任务在其时间窗口内选择服务时产生的零散“时间碎片”.因此,可以得出式(9)中任务 i 的最终最早开始时间 $T_{fes}(i)$ 与最终完成时间 $T_{ff}(i)$,其中 $S_{i\_best}.t$ 表示任务 i 选择最优服务的执行时间.

$$\begin{cases} T_{\text{fes}}(i) = \begin{cases} \beta_i, & pre(i) = \varnothing, \\ \max\limits_{j \in pre(i)} \{T_{\text{ff}}(j) + |e_{ji}|\}, & \text{otherwise}, \end{cases} \\ T_{\text{ff}}(i) = T_{\text{fes}}(i) + S_{i\_\text{best}}.t. \end{cases} \tag{9}$$

CACO 对 CA-DAG 表示的工作流调度时的伪码如下：

算法 2. CACO 调度阶段.

输入：工作流 G 及各任务的候选服务集合和用户截止期约束 δ；

输出：调度结果，即每个任务所选的最优服务.

① 按照 FC 计算 $T_{\min}$；

② 按照算法 1 进行分层，得到 T_{BLmin}；

③ if ($\delta < T_{\min}$)

④ 无法找到满足时间约束的调度方案；

⑤ else if ($T_{\min} \leqslant \delta < T_{\text{BLmin}}$)

⑥ 选取最快服务，计算整体执行费用 C；

⑦ else

⑧ 计算冗余时间 T_{LS}，按照式(8)确定各任务的时间窗口 $W1:[\beta_i, \delta_i]$；

⑨ 对各任务进行调度，调度过程中按照式(9)迭代确定各任务的开始时间 $T_{\text{fes}}(i)$ 及结束时间 $T_{\text{ff}}(i)$，选出各任务的最优服务，计算出整体费用 C；

⑩ endif

⑪ 返回调度结果.

3.3 复杂性分析

假定任务规模大小为 n，每个任务的服务规模大小为 m.

首先，采用 FC 规则求解任务的最快服务和工作流的最小完工时间时，需要针对每个任务遍历其所有服务计算其与该任务的所有前驱任务之间的通信开销，因此时间复杂度为 $O(mn)$.

在 CACO 调度阶段，算法 1 中行①和行⑤遍历所有任务计算他们的分层时间，在遍历的过程中需要检查任务的前驱或后继集合，在最坏的情况下时间复杂度为 $O(n^2)$. 算法 2 中行⑥和行⑨为各任务选取合适的服务，需要遍历各任务的服务集合，其中行⑨需重新计算任务的最早开始时间，要遍历任务的前驱集合，因此时间复杂度为 $O(mn)$；其他语句则为常数时间，通常情况下任务规模远远大于每个任务的服务规模，所以 CACO 的时间复杂度为 $O(n^2)$. 又知先前基于分层算法的时间复杂度均为 $O(n^2)$，因此 CACO 在引入通信开销的同时并没有增加算法的时间复杂度.

3.4 案例研究

本节以图 1 和表 1 给出的工作流和服务集合为例给出 CACO 的求解过程，假定用户给定的截止期 $\delta=26$. 求解过程如图 2 所示，$W2$ 为任务最终的时间窗口，即 $[T_{\text{fes}}(i), T_{\text{ff}}(i)]$.

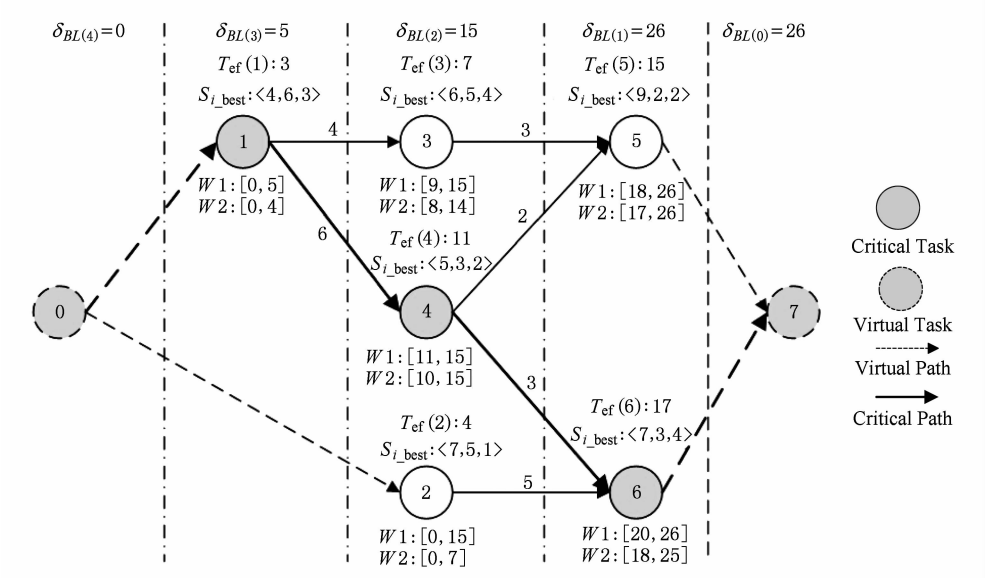


Fig. 2 The process of cost optimization of CACO.

图 2 CACO 费用优化过程示意图

以任务 5 为例，按照 FC 求解其最早完成时间，其前驱集合为{3,4}，服务集合如表 1 所示，遍历其所有服务求解选择此服务时的完成时间：max{7+3+3,11+2+3}=16，max{7+3+4,11+0+4}=15，max{7+3+7,11+2+7}=20，得到任务 5 的最早完成时间为 15，对应的服务为〈4,7,1〉. 可以看

到,按照 FC 求解出来的最快服务并不是服务集合中执行时间最短的服务,但却能够最快完成.最终按照 FC 求得的最小完工时间为 17,图 2 中粗箭头为关键路径.在分层阶段,可以得到任务 5 的时间窗口为〈18,26〉,时间窗口的大小为 8,最终分层结果如图 2 的虚线所示;在费用优化阶段,采用动态规划方法收集"时间碎片",因其前驱任务 3 在时刻 14 即完成,最终最早开始时间可修改为 14+3=17,时间窗口变为 9,因此可选择服务〈9,2,2〉.若在费用优化阶段不再收集"时间碎片",则得不到此最优服务.使用 CACO 求解,最终得到整个工作流的最优费用为 24.

4　实验方法

在仿真环境中实现了 CACO,DTL,DBL 和 TCDBL 这 4 种算法,并对这些算法的费用优化效果进行了评估.其中,DTL 和 DBL 分别是基于正向与逆向分层的工作流优化方法;TCDBL 是在逆向分层的基础上提出按并行度分配冗余时间,并基于任务的时序一致性进行调度,将具有同步特征的任务分配到同层进行费用优化.

4.1　实验设计

采用随机生成 DAG 图的方式,通过控制 DAG 图中任务的数目、服务集合大小、冗余时间比例和任务执行时间与通信时间的比例来产生大量的不同 DAG 实例.在本实验中,设置 DAG 工作流规模 $|V|\in[20,40,60,80,100]$,每种规模选取 10 个实例;服务集合规模 $|S(i)|\in[10,20,30,40,50]$;截止期 $\delta=T_{\min}+\theta\times T_{\min}$,其中冗余时间所占比例 $\theta\in[0.2,0.5,1,1.5,2]$;任务通信时间与执行时间的比例 $\varepsilon\in[0.2,0.4,0.6,0.8,1]$.综合上面各问题参数,共生成 7 500 个测试实例.

比较结果通过平均执行费用(average cost, AC)、最优解比例(optimal proportion, OP)及 CACO 相对于各算法的改进比(improvement ration of average cost, $IRAC$)进行衡量,性能指标为

$$AC(A)=\frac{\sum_{i=1}^{N}C_i(A)}{N},\tag{10}$$

$$OP(A)=\frac{N_B(A)}{N},\tag{11}$$

$$IRAC(A)=\frac{AC(A)-AC(\mathrm{CACO})}{AC(A)}.\tag{12}$$

其中,A 代表不同的算法,$C_i(A)$ 为第 i 个实例中 A 的目标函数值,N 为总的实验组数,$N_B(A)$ 为该组实验中 A 的目标函数值最小的实例个数,$AC(A)$ 为所有实例中 A 的平均执行费用.

4.2　前向一致规则验证

为了验证 FC 规则求解工作流最小完工时间的正确性,对比了在不同通信开销比例下使用 FC 和 MCP(不考虑通信开销和考虑通信开销这 2 种情况)求解出的最小完工时间.由图 3 可知,在通信开销为 0 时,FC 和 MCP 求解出相同的最小完工时间;加入通信开销后,FC 求解出的最小完工时间总位于 MCP 的 2 种情况之间,且随着通信开销比例逐渐增加,FC 与 MCP 这 2 种情况之间的差值也随之增加.

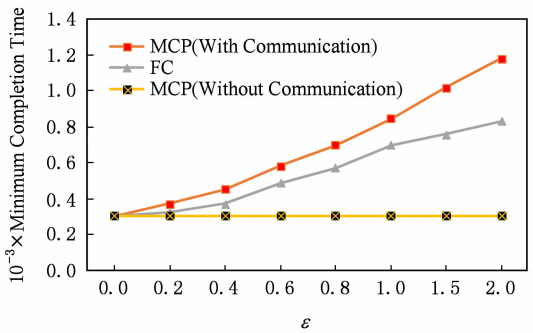


Fig. 3　The minimum completion time of workflow determines by FC and MCP.

图 3　FC,MCP 确定的工作流最小完工时间

4.3　不同问题参数下的费用优化效果比较

本文对比了 DTL,DBL,TCDBL,CACO 这 4 种算法在不同问题参数下(工作流规模、服务池规模、不同的冗余时间及通信开销所占比例)的费用优化效果.本文的定价机制为执行费用随着执行时间的增加线性下降.

4.3.1　不同工作流规模下的费用优化效果

图 4 为不同工作流规模下各算法的费用优化效果对比,可以看出,随着工作流规模的增加各算法的

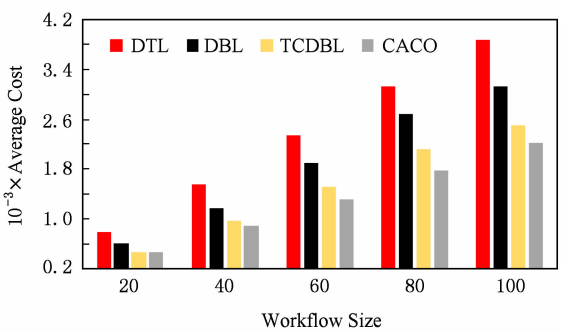


Fig. 4　The results of cost optimization at different workflow sizes.

图 4　不同工作流规模下的费用优化效果

执行费用都有所增长,CACO 的执行费用总能保持在最低,从而说明 CACO 可适用于任何规模的工作流;TCDBL 总是优于 DBL 和 DTL 算法,是因为其基于任务的时序一致性采取宽松的时间约束规则来选择服务,使得零散的"时间碎片"大大减少;从总体来看,CACO 的费用优化效果比 DTL, DBL, TCDBL 有显著提高.

4.3.2 不同服务规模下的费用优化效果

图 5 为不同服务规模下各算法费用优化效果的对比,服务规模为各任务服务集合的大小. 从图 5 可以看出,在任何服务规模下 CACO 总保持着最低的平均执行费用,这是因为 CACO 在任务调度过程中收集了每个任务执行完毕后产生的时间碎片,从而增大未调度任务的时间窗口,使任务的费用优化空间变大. 此外,DTL 总保持着最高的平均执行费用,算法性能较其他 3 个算法差很多.

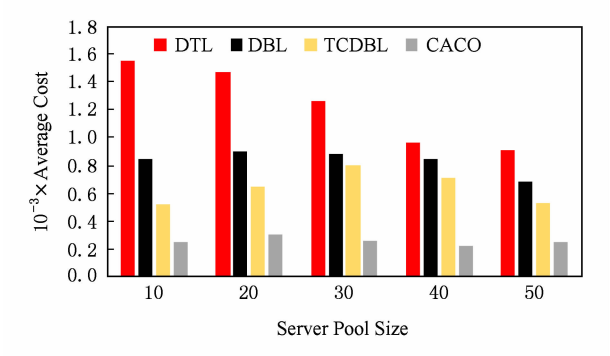


Fig. 5 The results of cost optimization at different server pool sizes.

图 5 不同服务规模下的费用优化效果

4.3.3 冗余时间对费用优化效果的影响

为探究冗余时间对算法的影响,本文对比了不同的冗余时间下各算法的费用优化效果. 从图 6 可以看出,随着冗余时间的增加各算法的优化空间逐渐增大,算法之间的费用优化差距在逐渐减小,最后差距为 0,此时所有算法为任务分配的都是费用最低的服务;此外,随着冗余时间的增加,CACO 的平均执行费用变化不大,总能最接近于最优费用,因此,在冗余时间很少时 CACO 有着很好的费用优化效果;TCDBL 先于 DTL,DBL 得到最低费用的服务,优于其他 2 种算法.

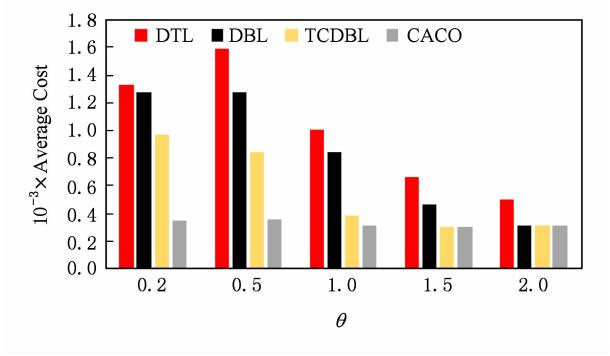


Fig. 6 The results of cost optimization at different superfluous time.

图 6 冗余时间对费用优化效果的影响

4.3.4 通信开销规模对费用优化效果的影响

图 7 为不同的通信开销规模下各算法的费用优化效果对比. 从图 7 可以看出,随着通信开销所占比例的逐渐增大,CACO 较其他算法的费用优化效果的提高也逐渐增大,这是因为其他算法都没有很好地处理通信开销,而 CACO 在调度的过程中采用动态规划的方法收集零散的"时间碎片",增大任务的优化空间;在通信开销比例很小的情况下,CACO 和 TCDBL 的费用优化效果相当,但都优于 DTL 和 DBL.

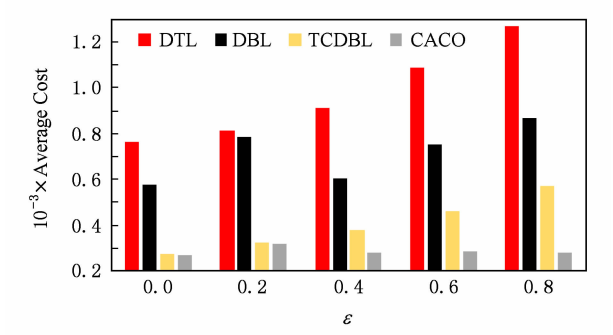


Fig. 7 The results of cost optimization at different communication overhead.

图 7 不同通信开销规模下的费用优化效果

4.4 所有问题参数下的算法比较

最后,本文在整体上对比了 4 种算法的平均执行费用 *AC*、最优解比例 *OP* 及 CACO 相对于各算法的改进比 *IRAC*. 结果如表 2 所示,当通信开销为 0 时,TCDBL 存在一定的最优解,但大体最优解都为 CACO,最终可以得出 CACO 的费用优化效果平均上比 DTL,DBL,TCDBL 分别提高了 59%,47%,29%.

Table 2 Comparison of the Algorithms at all Parameters

表 2 所有问题参数下各算法对比

Algorithm	*AC*	*OP*/%	*IRAC*
DTL	1 337	0	0.59
DBL	1 034	0	0.47
TCDBL	760	1	0.29
CACO	539	99	

从各种算法在不同参数设置下的平均执行费用来看,工作流规模越大通信开销所占比例越大,CACO 相对与其他算法的优势就越明显,这就表明 CACO 更加适用于规模较大、结构较复杂的工作流调度.这是因为,工作流规模越大、结构越复杂,那么工作流的层次会越复杂:1) CACO 在使用 FC 计算每个任务的最早开始执行时间时,不像其他算法选择执行时间最短的服务,而是选择执行可以最快结束的服务,这就避免了任务的没必要等待时间,使得任务执行更加紧凑;2)CACO 在调度完一个任务后,就会重新修订已调度任务的结束时间和未调度任务的最早开始时间,相较于其他调度算法可以更有效地收集"时间碎片",将未调度任务的时间窗口进一步扩大,从而选择出费用更低的服务.

5　结　论

考虑到现有工作流费用优化模型缺乏对通信开销的考虑这个问题,本文设计了带有通信开销的工作流费用优化模型,使其适用于真实的云环境.基于分层算法对任务的执行费用进行优化,提出算法 CACO,采用的 FC 能准确地求出工作流的最小完工时间;在算法的调度过程中采用动态规划方法收集零散的"时间碎片",增大任务的费用优化空间.经过与算法 DTL,DBL,TCDBL 的对比模拟实验,证明了在考虑通信开销的前提下算法 CACO 的费用优化效果比算法 DTL,DBL,TCDBL 都有显著提高.

参　考　文　献

[1] Vaquero L M, Rodero M L, Caceres J, et al. A break in the clouds: Towards a cloud definition [J]. ACM SIGCOMM Computer Communication Review, 2008, 39(1): 50-55

[2] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Bottom level based heuristic for workflow scheduling in grids [J]. Chinese Journal of Computers, 2008, 31(2): 283-290 (in Chinese)
(苑迎春, 李小平, 王茜, 等. 基于逆向分层的网格工作流调度算法[J]. 计算机学报, 2008, 31(2): 282-290)

[3] Yuan Yingchun, Li Xiaoping, Wang Qian. Cost optimization heuristics for grid workflows scheduling based on serial reduction [J]. Journal of Computer Research and Development, 2008, 45(2): 246-253 (in Chinese)
(苑迎春, 李小平, 王茜. 基于串规约的网格工作流费用优化方法[J]. 计算机研究与发展, 2008, 45(2): 246-253)

[4] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Deadline division-based heuristic for cost optimization in workflow scheduling [J]. Information Sciences, 2009, 179(15): 2562-2575

[5] Liu Cancan, Zhang Weimin, Luo Zhigang, et al. Temporal consistency based heuristics for cost optimization in workflow scheduling [J]. Journal of Computer Research and Development, 2012, 49(6): 1323-1331 (in Chinese)
(刘灿灿, 张卫民, 骆志刚, 等. 基于时序一致的工作流费用优化方法[J]. 计算机研究与发展, 2012, 49(6): 1323-1331)

[6] Liu Cancan, Zhang Weimin, Luo Zhigang. Path balance based heuristics for cost optimization in workflow scheduling [J]. Journal of Software, 2013, 24(6): 1207-1221 (in Chinese)
(刘灿灿, 张卫民, 骆志刚. 基于路径平衡的工作流费用优化方法[J]. 软件学报, 2013, 24(6): 1207-1221)

[7] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Grid workflows schedule based on priority rules [J]. Acta Electronica Sinica, 2009, 37(7): 1457-1464 (in Chinese)
(苑迎春, 李小平, 王茜, 等. 基于优先级规则的网格工作流调度[J]. 电子学报. 2009, 37(7): 1457-1464)

[8] Liu Cancan, Zhang Weimin, Luo Zhigang, et al. Workflow cost optimization heuristics based on advanced priority rule [J]. Journal of Computer Research and Development, 2012, 49(7): 1593-1600 (in Chinese)
(刘灿灿, 张卫民, 骆志刚, 等. 基于改进优先级规则的工作流费用优化方法[J]. 计算机研究与发展, 2012, 49(7): 1593-1600)

[9] Ke Hua, Ma Weimin, Ni Yaodong. Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem [J]. Applied Mathematics and Computation, 2009, 215(1): 308-313

[10] Chen Weineng, Zhang Jun. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements [J]. IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2009, 39(1): 29-43

[11] Hunold S, Rauber T, Suter F. Scheduling dynamic workflows onto clusters of clusters using postponing [C] //Proc of the 8th IEEE Int Symp on Cluster Computing and the Grid. Piscataway, NJ: IEEE, 2008: 669-674

[12] Chen Shiping, Nepal S, Liu Ren. Secure connectivity for intra-cloud and inter-cloud communication [C] //Proc of the 40th Int Conf on Parallel Processing Workshops (ICPPW 2011). Piscataway, NJ: IEEE, 2011: 154-159

[13] N'takpé T, Suter F, Casanova H. A comparison of scheduling approaches for mixed-parallel applications on heterogeneous platforms [C] //Proc of the 6th Int Symp on Parallel and Distributed Computing (ISPDC'07). Piscataway, NJ: IEEE, 2007: 250-257

[14] Hönig U, Schiffmann W. A meta-algorithm for scheduling multiple dags in homogeneous system environments [C] //Proc of the 18th Int Conf on Parallel and Distributed Computing and Systems (PDCS'06). Calgary, AB, Canada: ACTA Press, 2006: 147-152

[15] Zhao Henan, Sakellariou R. Scheduling multiple DAGs onto heterogeneous systems [C] //Proc of the 20th IEEE Int Parallel and Distributed Processing Symp (IPDPS'06). Piscataway, NJ: IEEE, 2006: 130-143

[16] Stanzani S L, Sato L M, Netto M A S. Scheduling workflows in multi-cluster environments [C] //Proc of the 27th Int Conf on Advanced Information Networking and Applications Workshops (WAINA'13). Piscataway, NJ: IEEE, 2013: 560-565

[17] Topcuoglu H, Hariri S, Wu Minyou. Performance-effective and low-complexity task scheduling for heterogeneous computing [J]. IEEE Trans on Parallel and Distributed Systems, 2002, 13(3): 260-274

[18] Choudhary V, Choudhury T, Kacker S, et al. An approach to improve task scheduling in a decentralized cloud computing environment [J]. International Journal of Computer Technology & Applications, 2012, 3(1): 312-316

[19] Gao Yue, Wang Yanzhi, Gupta S K, et al. QoS-based scheduling of workflow applications on service grids, GRIDS-TR-2005-8 [R]. Melbourne, Victoria, Australia: The University of Melbourne, Grid Computing and Distributed Systems Laboratory, 2005

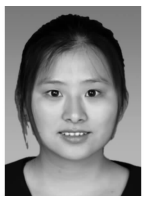
[20] Kumar B A, Ravichandran T. Time and cost optimization algorithm for scheduling multiple workflows in hybrid clouds [J]. European Journal of Scientific Research, 2012, 89(2): 265-275

[21] Gao Yue, Wang Yanzhi, Gupta S K, et al. An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems [C] //Proc of the 9th IEEE/ACM/IFIP Int Conf on Hardware/Software Codesign and System Synthesis. Piscataway, NJ: IEEE, 2013: 31-40

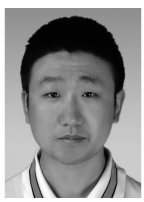
[22] Yu Jia, Buyya R, Tham C K. Cost-based scheduling of scientific workflow applications on utility grids [C] //Proc of the 1st IEEE Int Conf on e-Science and Grid Computing (e-Science'05). Piscataway, NJ: IEEE, 2005: 140-147

[23] Du Jun, Yu Ce, Sun Jizhou, et al. EasyHPS: A multilevel hybrid parallel system for dynamic programming [C] //Proc of the 27th IEEE Int Symp on Parallel & Distributed Processing Workshops and PhD Forum (IPDPSW'13). Piscataway, NJ: IEEE, 2013: 630-639

Guo He, born in 1955. PhD and professor in Dalian University of Technology. Member of China Computer Federation. His main research interests includes high performance computing, computer architecture, parallel computing (guohe@dlut.edu.cn).



Chen Zheng, born in 1989. Master. Her main research interests include cloud computing, computer architecture and distributed computing (chenz1989@163.com).



Yu Yulong, born in 1985. PhD candidate. Student member of China Computer Federation. His main research interests include computer architecture and parallel computing (yuylong@acm.org).

Wang Yuxin, born in 1973. PhD and associate professor in Dalian University of Technology. Member of China Computer Federation. His main research interests include computer architecture, parallel computing (wyx@dlut.edu.cn).

Chen Xin, born in 1983. PhD and lecturer in Dalian University of Technology. His main research interests include combination optimization, scheduling problem (cx.dlut@gmail.com).