

基于异构云计算的成本约束下的工作流能量高效调度算法



张龙信 周立前 文 鸿 肖满生 邓晓军

湖南工业大学计算机学院 湖南 株洲 412007

(longxinzhang@hut.edu.cn)

摘 要 云计算已成为各行业中十分重要的计算服务方式。传统的云计算研究主要侧重于云服务的定价方式、利润最大化、执行效率等服务质量,而绿色计算成为了近年来分布式计算的发展趋势。针对异构云环境中满足云用户计算成本约束的工作流任务集调度问题,提出了一种低时间复杂度、能量感知的预算等级调度(Energy-Aware Based on Budget Level Scheduling, EABL)算法。EABL算法包含并行任务集任务优先级的建立、任务预算成本的分配及最优执行虚拟机和能量高效频率的确定3个主要阶段,能在满足预算成本约束的同时最大限度地降低任务集执行过程中的能量消耗。采用真实世界的大规模工作流任务集对算法进行测试,结果表明,与著名的EA_HBCS和MECABP算法相比,EABL算法在充分利用预算成本的同时,有效地降低了工作流任务集在云数据中心计算过程中的能量消耗。

关键词: 异构云;工作流调度;成本约束;能量高效;任务调度

中图法分类号 TP301

Energy Efficient Scheduling Algorithm of Workflows with Cost Constraint in Heterogeneous Cloud Computing Systems

ZHANG Long-xin, ZHOU Li-qian, WEN Hong, XIAO Man-sheng and DENG Xiao-jun

School of Computer Science, Hunan University of Technology, Zhuzhou, Hunan 412007, China

Abstract Cloud computing has become a very important computing service mode in various industries. Traditional studies on cloud computing mainly focus on the research of service quality such as the pricing mode, profit maximization and execution efficiency of cloud services. Green computing is the development trend of distributed computing. Aiming at the scheduling problem of workflow task set that meets the computing cost constraint of cloud users in heterogeneous cloud environment, an energy-aware based on budget level scheduling algorithm(EABL) with low time complexity is proposed. The EABL algorithm consists of three main stages; task priority establishment, task budget cost allocation, optimal execution virtual machine and energy efficiency frequency selection of the parallel task set, so as to minimize the energy consumption during task set execution under the constraint of budget cost. A large-scale workflow task sets in the real world are used to conduct a large number of tests on the algorithm for the experiment in this paper. Compared with famous algorithms EA_HBCS and MECABP, EABL algorithm can effectively reduce the energy consumption in the computing process of cloud data centers by making full use of the budget cost.

Keywords Heterogeneous computing, Workflow scheduling, Budget constraint, Energy efficiency, Task scheduling

1 引言

绿色计算是计算机科学的一个最新趋势,它试图减少集群、网格和云等分布式平台上的计算机所产生的能源消耗,从而实现低碳计算。传统的调度解决方案在不考虑能量成本的情况下最小化任务集的处理时间,通常设计合理的调度策略来减少能量消耗,在特定的资源上分配任务,从而缩短处理时间并节约能量。

云服务需求的不断增长,对大型云数据中心的计算、存储等资源的分配和利用提出了更高的要求。数据中心虚拟机的计算和迁移等操作采用基于欧氏距离的多目标虚拟机资源分配方法,可以节约能量开销,最大限度地减少资源的浪费,同时避免了云数据中心的等级协议(Service Level Agreement, SLA)的冲突^[1]。云中心和云服务中的绿色计算问题已经引起了国内外研究人员的广泛关注。

云服务已被广泛应用于社会生活的各个领域,云用户和

到稿日期:2020-03-09 返修日期:2020-06-29 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家重点研发计划(2018YFB1003401);国家自然科学基金(61702178,61672224);湖南省自然科学基金(2019JJ50123,2020JJ6087,2019JJ60054,2018JJ4068);中国国家留学基金(201808430297)

This work was supported by the National Key R&D Program of China(2018YFB1003401), National Natural Science Foundation of China(61702178,61672224), Natural Science Foundation of Hunan Province(2019JJ50123,2020JJ6087,2019JJ60054,2018JJ4068) and China Scholarship Council(201808430297).

通信作者:文鸿(wenhut@163.com)

云供应商之间通过 SLA 协定双方的需求和服务质量。异构云计算系统中,除了满足云用户提交的作业调度效率外,如何降低云系统的能量消耗也是一个异常重要和持久的话题。已有研究主要关注云服务的定价策略、云计算任务的调度长度最小化问题以及云计算中心效益最大化等问题。因此,本文针对异构云计算环境中成本预算约束下的工作流任务集,提出能量感知的高效调度算法,所提 EABL 算法在保证服务质量的前提下降低了能量消耗。在 EABL 算法中,首先建立工作流任务集的调度优先级;然后确定每个任务的预算成本,以便其获得较合理的计算成本并为其选择虚拟机;最后根据任务的最终预算成本,快速地选择合适的虚拟机和最佳执行频率,完成任务的映射,最大限度地节省能量。

2 相关工作

近年来,在云计算领域学者们取得了许多优秀的成果。针对业界广泛接受的考虑通信竞争的任务调度模型,Zhang 等^[2]提出截止时间和能量预算双重约束下并行任务集的可靠性感知的调度算法,该算法包含任务的优先级计算、通信边的调度以及松弛时间再利用 3 个阶段,进一步提高了任务集执行的可靠性。Mansouri 等^[3]针对云用户中数据的存储和迁移,采用动态规划和线性规划技术提出的最佳离线算法,有效节省了数据存储和迁移中的成本开销。Li^[4]针对云计算系统中多核处理器上具有连续或离散速度级的优先约束并行任务的能量和时间约束调度,提出了能耗预分配的数学模型,其提升了并行任务集高效调度的性能。Zhang 等^[5]针对云无线网中高能效和低碳的需求,提出了一种联合优化方法,使得云无线系统在用户访问、资源分配、负载均衡方面更加灵活。Zhang 等^[6]针对异构云计算系统中能量预算约束下的并行任务集,提出了任务集可靠性最大化的优化策略。Juarez 等^[7]提出了一个实时动态调度系统,在分布式计算平台上高效地执行基于任务的应用程序,减少了能源消耗。

鉴于异构云系统中并行任务集调度过程中低能量消耗和高系统可靠性两个重要指标的对立关系,Zhang 等^[8]提出了一种新的双目标遗传算法,当用户提交作业至云数据中心时,该策略可为用户提供针对自身需求的多样化解决方案。在新兴的雾计算环境中,Shojafar 等^[9]针对小型自供电数据中心(如雾节点)中分布在远程云与车辆客户端之间的实时数据传播服务,提出了一种高效、节能的自适应资源调度器,极大地提高了系统的能源利用效率。云计算与移动网络相结合,产生了移动云计算。为了缓解移动设备资源约束的限制,提升网络服务质量和效率,Peng 等^[10]提出了动态电压频率调节技术和鲸鱼优化算法的移动设备最优任务工作流调度方案,来解决工作流任务集截止时间和能量消耗的联合优化问题。Li 等^[11]针对截止时间约束的工作流任务集,提出了成本和能量感知的算法,以最小化工作流的执行成本并降低能耗。

Abrishami 等^[12]针对云环境中的基础设施即服务的工作流关键路径的特征,在满足服务质量和截止期限的前提下,设计了大型工作流的调度策略,降低了工作流的执行成本。针对预算约束的工作流任务集,Arabnejad 等^[13]根据任务的时效率和计算开销成本确定预算等级,并设计了著名的异构预

算约束调度算法(Heterogeneous Budget Constrained Scheduling, HBCS),在满足预算成本约束的前提下,该算法实现了工作流的高效调度;Chen 等^[14]使用可用的成本开销预分配策略,较好地解决了异构云环境中成本约束的并行任务集调度问题。然而,如何进一步降低预算下的工作流任务集的能量,提升算法的效率,依然是研究的难点。

3 异构云系统的相关知识

3.1 系统模型

本文的目标系统为异构分布式云计算系统,系统中的计算节点分布在由计算、存储、网络带宽各异的节点构成的资源池中。用户提交的作业可以分解为多个任务,每个任务可按照一定的调度策略分配至从资源池虚拟化后的虚拟机上执行。虚拟机节点集用 $VM = \{vm_0, vm_1, vm_2, \dots, vm_m\}$ 表示。系统中的各虚拟机之间通过高速的互联网络相连,网络延迟低。假定用户提交的作业均为计算规模较大的工作流集,各虚拟机启动开销可以忽略不计。虚拟机中的处理器支持多档离散的运行频率。表 1 列出了 3 种虚拟机中处理器支持的运行频率^[6],该频率是归一化后的结果,即支持的频率与支持的最高频率的比值。

表 1 虚拟机中处理器支持的运行频率

Table 1 Running frequency supported by processor in VMs

Frequency levels	vm_0	vm_1	vm_2
0	1.00	1.00	1.00
1	0.80	0.90	0.85
2	0.70	0.80	0.65
3	0.60	0.70	0.50
4	0.50	0.60	0.35
5	0.40	0.50	—
6	—	0.40	—

3.2 任务模型

典型的工作流任务集通常可以用有向无环图(Direct Acyclic Graph,DAG)来描述,表示为 $G = \{T, E, C, W\}$ 。其中, T 为任务节点集合, $T = \{\tau_0, \tau_1, \tau_2, \dots, \tau_n\}$; E 为 DAG 的有向边集合; C 为有优先约束关系的任务之间通信开销的集合, W 为任务节点在不同虚拟机上的计算开销集合。对于 T 中的任务 τ_i , 其所有的直接父亲节点被称为前驱节点,记为 $parent(\tau_i)$; τ_i 的所有直接孩子节点被称为后继节点,记为 $child(\tau_i)$ 。若 $parent(\tau_i) = \emptyset$, 则称 τ_i 为入口节点,记作 τ_{entry} ; 若 $child(\tau_i) = \emptyset$, 则称 τ_i 为出口节点,记作 τ_{exit} 。DAG 中的通信边 $e_{i,j} \in E$, 表示 τ_i 为 τ_j 的直接父亲节点,当且仅当 τ_i 执行完成后,任务 τ_j 才有可能准备就绪,等待执行。DAG 中的通信开销 $c_{i,j} \in C$, 表示当任务 τ_i 和 τ_j 不在同一个虚拟机上执行时,数据从任务 τ_i 所在虚拟机全部传输至任务 τ_j 所在的虚拟机所产生的通信开销。若任务 τ_i 和 τ_j 在同一虚拟机上执行,此通信开销可忽略不计,记作零。任务在虚拟机上的计算开销 $w_{i,j} \in W$, 表示任务 τ_i 在虚拟机 vm_j 上执行时所需要的计算开销。本文约定,对于一个给定的工作流图,有且仅有一个入口节点和一个出口节点。当 DAG 图不满足此条件时,可以通过增加零计算开销的伪节点和零通信开销的通信边与相邻的子节点和父节点连接,从而构成虚拟的入口节点或者出口节点。一个典型的工作流图如图 1 所示,其中共包含 11 个

任务节点。以任务节点 τ_3 为例,其父亲节点为 τ_0 ,其后继节点为 $\{\tau_5, \tau_6, \tau_7\}$ 。 τ_0 和 τ_{10} 分别为此工作流图的入口节点和出口节点。

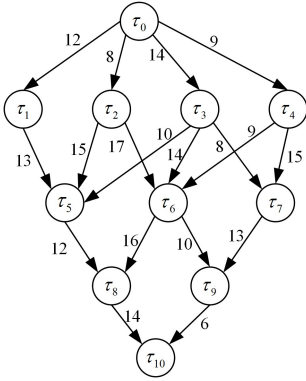


图1 一个典型的工作流图

Fig. 1 Typical workflow

图1中典型的工作流图在各虚拟机上的计算开销如表2所列。以任务 τ_5 为例, τ_5 在虚拟机 vm_0 上的计算开销为18, 记为 $w_{5,0}=18$ 。图1中,当 τ_5 和其前驱节点 τ_2 被分配至两个不同的虚拟机上执行时,数据从 τ_2 传输至 τ_5 的通信开销为15,记为 $c_{2,5}=15$ 。

表2 图1中任务在各虚拟机上的计算开销

Table 2 Computational cost of tasks on each virtual machine in Fig. 1

Task _i	vm ₀	vm ₁	vm ₂
τ_0	15	11	10
τ_1	13	10	11
τ_2	12	9	7
τ_3	12	17	8
τ_4	10	18	16
τ_5	18	13	12
τ_6	14	15	13
τ_7	8	16	9
τ_8	16	14	7
τ_9	11	12	10
τ_{10}	12	9	11

3.3 能耗模型

云计算资源池中各处理器的能量消耗采用能耗模型进行刻画,处理器的功率计算如式(1)所示:

$$P = P_s + h(P_{ind} + P_d) \\ = P_s + h(P_{ind} + C_{eff} f^a) \quad (1)$$

其中,处理器的功耗包含静态功耗和动态功耗两部分,静态功耗用 P_s 表示,动态功耗又包含与处理器运行频率有关的动态功耗 P_{ind} 以及与处理器运行频率无关的动态功耗 P_d 两部分; h 的取值为0或者1; P_d 可以表示为 $C_{eff} f^a$, C_{eff} 为有效切换电容参数, f 为处理器运行的频率, a 为常数; P_{ind} 为服从均匀分布的常数。由于静态功耗相对于动态功耗来说通常较小,因此任务在虚拟机上以频率运行时的能耗如式(2)所示:

$$E_i(f_i) = P_{ind} \cdot \frac{c_i}{f_i} + c_{eff} \cdot c_i \cdot f_i^a \quad (2)$$

其中, c_i 为任务 τ_i 的计算开销, f_i 为执行任务 τ_i 选择的运行频率。

定义1(任务的能量高效频率 f_{ee}) 任务在选择的虚拟

上执行时,存在一个能量高效频率 f_{ee} ,使得任务在该虚拟机上以此频率执行时消耗的能量最少。 f_{ee} 的计算表达式如式(3)所示:

$$f_{ee} = \sqrt{\frac{a}{(a-1)C_{eff}} \frac{P_{ind}}{c_i}} \quad (3)$$

对式(2)求二阶导,得到 $E_i''(f_i) > 0$ 。这表明 $E_i(f_i)$ 是一个凹函数,即存在一个驻点,使得 $E_i(f_i)$ 取得极小值。当且仅当 $E_i(f_i)$ 的一阶导数 $E_i'(f_i) = 0$ 时, $E_i(f_i)$ 取得最小值。

$$E_i'(f_i) = -P_{ind} \cdot \frac{c_i}{f_i^2} + a \times c_{eff} \cdot c_i \cdot f_i^{a-1} = 0 \quad (4)$$

对式(4)进行化简,即可得到式(3)。

定义2(工作流任务集的能耗 E_{total}) 工作流集的总能量消耗等于各任务的能量开销之和,如式(5)所示:

$$E_{total}(G) = \sum_{i=1}^n E_i(f_i) \quad (5)$$

3.4 成本模型

为了方便描述,下文给出文中使用的成本计算的相关概念的定义。

定义3(任务向上优先级 $URank$) 工作流任务集中任务的 $URank$ 自底向上,递归计算任务的向上优先级。任务 τ_i 的 $URank(\tau_i)$ 计算表达式如式(6)所示:

$$URank(\tau_i) = \max_{\tau_k \in child(\tau_i)} \{URank(\tau_k) + c_{i,k}\} + \bar{w}_i \quad (6)$$

其中, $\bar{w}_i$ 为 τ_i 的后继节点在各虚拟机上执行的计算开销的平均值。

定义4(任务向下优先级 $DRank$) 工作流任务集中任务的向下优先级 $DRank$ 自顶向下递归计算。任务 τ_i 的 $DRank(\tau_i)$ 的计算表达式如式(6)所示:

$$DRank(\tau_i) = \max_{\tau_k \in parent(\tau_i)} \{DRank(\tau_k) + c_{k,i} + \bar{w}_k\} \quad (7)$$

定义5(任务实际的计算开销) 任务 τ_i 在虚拟机 vm_j 上以频率 $f_{j,k}$ 执行时,所需要的计算开销如式(8)所示:

$$cost(\tau_i, vm_j, f_{j,k}) = w_{i,j} \times \frac{f_{j,max}}{f_{j,k}} \times p(vm_k) \quad (8)$$

其中, $f_{j,max}$ 是虚拟机 vm_j 上归一化的最大频率, $p(vm_k)$ 为虚拟机 vm_j 的单位计算成本。

定义6(工作流集的计算开销) 工作流集 G 的计算开销等于该任务集内所有的任务节点计算开销的总和,其计算表达式如式(9)所示:

$$cost(G) = \sum_{i=1}^{|N|} cost(\tau_i) = \sum_{i=1}^{|N|} cost(\tau_i, vm(p_i), f(p_i)) \quad (9)$$

其中, $|N|$ 为任务集中节点的数量, $vm(p_i)$ 为运行任务所在的虚拟机, $f(p_i)$ 为任务执行时选择的频率。

定义7(任务 τ_i 的最小计算开销) 由式(8)可知,当 τ_i 在指定的虚拟机上执行时,以最大归一化频率执行的计算开销最小。因此,遍历所有虚拟机,分别以最大归一化后的频率在各虚拟机上执行 τ_i 时,得到的结果中的最小值便为任务 τ_i 的最小计算开销。任务 τ_i 的最小计算开销的表达式如式(10)所示:

$$cost_{min}(\tau_i) = \min_{vm_k \in VM} \{cost(\tau_i, vm_k, f_{k,max})\} \quad (10)$$

定义8(任务 τ_i 的最大计算开销) 当 τ_i 在指定的虚拟机上执行时,以最小归一化频率执行的计算开销值最大。任

务 τ_i 的最大计算开销的计算表达式如式(11)所示:

$$cost_{\max}(\tau_i) = \min_{vm_k \in VM} \{cost(\tau_i, vm_k, f_{k, \min})\} \quad (11)$$

定义 9(工作流集 G 的最小计算成本) 工作流集中的最小计算成本等于任务集中各任务的最小计算成本开销之和,其表达式如式(12)所示:

$$cost_{\min}(G) = \sum_{i=1}^{|N|} cost_{\min}(\tau_i) \quad (12)$$

定义 10(工作流集 G 的最大计算成本) 工作流集中的最大计算成本等于任务集中各任务的最大计算成本开销之和,其表达式如式(13)所示:

$$cost_{\max}(G) = \sum_{i=1}^{|N|} cost_{\max}(\tau_i) \quad (13)$$

定义 11(工作流集的 G 松弛成本 $SC(G)$) 工作流集的松弛成本为工作流集的总预算成本 $cost_{\text{bgt}}(G)$ 与工作流集最小计算成本 $cost_{\min}(G)$ 之差,其计算表达式如式(14)所示:

$$SC(G) = cost_{\text{bgt}}(G) - cost_{\min}(G) \quad (14)$$

定义 12(任务的预算等级) 任务的预算等级等于其向上和向下优先级之和与整个任务集中所有任务向上、向下优先级之和的比值,其计算表达式如式(15)所示:

$$BL(\tau_i) = \frac{URank(\tau_i) + DRank(\tau_i)}{\sum_{j=1}^{|N|} \{URank(\tau_j) + DRank(\tau_j)\}} \quad (15)$$

预算等级表示当前任务在任务集中的优先程度,该值越大,表示可从任务集的松弛成本中获得更多的松弛成本。

定义 13(任务的预算等级成本) 任务的预算等级成本 $cost_{bl}(\tau_i)$ 等于其最小计算成本加上其对应的预算等级与松弛成本之积,其计算表达式如式(16)所示:

$$cost_{bl}(\tau_i) = cost_{\min}(\tau_i) + SC \times BL(\tau_i) \quad (16)$$

定义 14(任务的预算成本) 工作流中的任务在调度的过程中,前一个任务实际产生的计算开销可能小于其预算的成本,该任务剩余的成本部分可以累积给准备就绪任务,以便其有更加宽裕的预算来选择能量最优频率。任务的预算成本 $cost(\tau_n)$ 的计算表达式如式(17)所示:

$$cost(\tau_n) = cost_{\text{bgt}}(G) - \sum_{j=1}^{n-1} cost(\tau_j) - \sum_{k=n+1}^{|N|} cost_{bl}(\tau_s(k)) \quad (17)$$

其中, $\sum_{j=1}^{n-1} cost(\tau_j)$ 表示已经完成调度的任务的实际计算开销,

$\sum_{k=n+1}^{|N|} cost_{bl}(\tau_s(k))$ 为尚未调度任务的预算等级成本。

3.5 问题描述

绿色计算是分布式计算中的一个十分重要的课题。在异构云计算系统中,对于用户提交的工作流任务集 G ,云服务中心在满足用户需求且不超过用户预算的前提下,最大限度地降低任务执行过程的能量消耗 $E_{\text{total}}(G)$,是本文将要探索解决的科学问题。该问题的数学形式化表示如式(18)和式(19)所示:

$$\text{Minimize: } E_{\text{total}}(G) \quad (18)$$

$$\text{subject to: } cost(G) = \sum_{i=1}^{|N|} cost(\tau_i, vm_{\text{map}}(\tau_i)) \leq cost_{\text{bgt}}(G) \quad (19)$$

其中, $vm_{\text{map}}(\tau_i)$ 为执行任务的虚拟机, $cost_{\text{bgt}}(G)$ 为任务集的总预算成本。

4 基于预算等级的能量高效算法

异构云计算系统中的工作流任务调度,可以归为组合优化问题。因为在多项式时间内求解组合优化问题得不到最优解,所以通常采用高效的启发式解决方案寻求次优解。为此,本文针对异构云计算系统中预算约束下的工作流任务的能量节约问题,提出了预算约束的能量感知算法 EABL。EABL 算法主要分为以下 3 个阶段:1)建立任务的优先约束队列,保证任务的调度不违反任务集的拓扑结构;2)计算工作流任务集的最小计算开销、最大计算开销以及任务的预算等级,最大限度地保证每个任务在调度过程中有足够的预算成本来选择对应的能量高效执行频率;3)根据任务的预算成本,快速确定最优的执行虚拟机和相应的执行频率,确保在当前的预算成本下消耗的能量最少。

4.1 EABL 算法

基于预算等级的能量感知调度算法的伪代码如算法 1 所示。步骤 1 计算工作流图中每个节点的 $URank$ 值;步骤 2 计算对应的 $DRank$;步骤 3 计算每个任务的最小计算开销 $cost_{\min}(\tau_i)$ 和最大计算开销 $cost_{\max}(\tau_i)$;步骤 4 计算任务的预算等级;步骤 5 计算每个任务的预算等级计算开销;步骤 6 计算工作流图的最小计算开销和最大计算开销;步骤 7 根据工作流图中任务的 $URank$ 值大小顺序建立任务集的调度优先等级,以保证任务集的优先约束关系。步骤 8—步骤 22 从优先队列 $Q_r(G)$ 中取出队头任务,计算其对应的预算成本,并为其选择该预算成本下最优的执行虚拟机及能量高效执行频率。其中,步骤 9 取出 $Q_r(G)$ 中的候选任务节点;步骤 10 根据当前工作流任务集的松弛程度重新计算候选节点的预算成本;步骤 12—步骤 20 为候选任务节点选择最佳执行的虚拟机,并在步骤 13 调用算法 2 根据候选任务的预算成本选择能量高效执行频率;步骤 21 更新工作流图的实际计算开销并移除 $Q_r(G)$ 中的队头节点,为下一个数据准备就绪的任务调度做好准备。当所有的任务调度完后,步骤 23 计算整个工作流图的实际成本开销,步骤 24 统计工作流图的能量消耗。

算法 1 基于预算等级的能量感知调度算法

输入:工作流任务集 $G = \{T, E, W, C\}$, $VM, cost_{\text{bgt}}(G)$,

虚拟机单位计算开销表

输出: $cost(G), E(G)$

1. 递归计算工作流图中每个节点的 $URank$ 值;
2. 递归计算工作流图中每个节点的 $DRank$ 值;
3. 计算任务节点的 $cost_{\min}(\tau_i)$ 和 $cost_{\max}(\tau_i)$;
4. 计算任务节点的 $BL(\tau_i)$;
5. 计算节点的 $cost_{BL}(\tau_i)$;
6. 计算工作流图的 $cost_{\min}(G)$ 和 $cost_{\max}(G)$;
7. 工作流图 G 根据节点的 $URank$ 值降序排列,得到任务优先队列 $Q_r(G)$;
8. while 队列 $Q_r(G)$ 非空
9. τ_{tmp} = 队头节点
10. 计算节点 τ_{tmp} 的预算开销 $cost_{\text{bgt}}(\tau_{\text{tmp}})$ 值
11. $E(\tau_{\text{tmp}}) \leftarrow +\infty$
12. for $j=0$ to $|V|$ do
13. 调用算法 2 获得 τ_{tmp} 在 vm_j 上的能量高效频率 $f_{\text{ef}}(\tau_{\text{tmp}})$

```
14.      计算  $\tau_{\text{tmp}}$  在  $\text{vm}_{j, f_{\text{ef}}}$  上的  $\text{cost}(\tau_{\text{tmp}}, \text{vm}_{j, f_{\text{ef}}})$ 
15.      计算  $\tau_{\text{tmp}}$  在  $\text{vm}_{j, f_{\text{ef}}}$  上的能量开销  $E(\tau_{\text{tmp}}, \text{vm}_{j, f_{\text{ef}}})$ 
16.      if  $E(\tau_{\text{tmp}}, \text{vm}_{j, k}) < E(\tau_{\text{tmp}})$ 
17.           $E(\tau_{\text{tmp}}) \leftarrow E(\tau_{\text{tmp}}, \text{vm}_{j, k})$ 
18.      标记任务计算开销
19.  end if
20. end for
21. 更新 G 的实际计算开销并移除队头节点
22. end while
23. 计算工作流图的实际成本开销  $\text{cost}(G)$ ;
24. 计算工作流图的  $E(G)$ .
```

4.2 搜索能量高效频率算法

搜索高效调度算法的伪代码如算法 2 所示。步骤 1 获取当前虚拟机(vm_k)的理想能量高效频率 f_{ee} 。步骤 2 计算当前任务 τ_i 在此虚拟机上以 f_{ee} 运行时所需要的计算开销。步骤 3—步骤 20 根据当前任务的预算成本大小,选择在当前虚拟机上执行时的最优执行频率,确保消耗的能量最小,并将此最优频率输出。其中,步骤 3—步骤 11 中对应当前任务的 $\text{cost}_{\text{bgt}}(\tau_i)$ 大于在 vm_k 上以 f_{ee} 执行时的计算开销,此时将有足够的成本选择能量高效执行频率 f_{ee} 。 f_{ee} 是一个理想频率, vm_k 并不一定支持这档频率,即可能 $f_{\text{ee}}(\text{vm}_k) \notin F(\text{vm}_k)$, 根据式(3)中的分析,此时可选的其他能量高效频率值一定在 f_{ee} 附近,因此只需从 f_{ee} 开始,往 vm_k 的 f_{min} 方向遍历,找到的第一频率即为能量高效最优频率。步骤 12—步骤 19 对应当前任务的 $\text{cost}_{\text{bgt}}(\tau_i)$ 值小于其在虚拟机 vm_k 上以频率 f_{ee} 执行时所需计算开销时的场景,此场景下的能量高效最优频率在 f_{ee} 附近, vm_k 自 f_{ee} 往 f_{max} 方向遍历该虚拟机支持的频率时,第一个满足步骤 14 中条件的频率即为能量高效频率。

算法 2 搜索能量高效频率

输入: $\text{vm}_k, \text{Cost}_{\text{bgt}}(\tau_i)$
输出: 任务 τ_i 在虚拟机 vm_k 上的能量高效频率 $f_{\text{ef}}(\tau_i)$

```
1. 获取当前虚拟机的能量最优频率  $f_{\text{ee}}(\text{vm}_k)$ ;
2. 计算最优频率下的开销  $\text{cost}(\tau_i, f_{\text{ee}}(\text{vm}_k))$ ;
3. if  $\text{cost}(\tau_i, \text{vm}_j, f_{\text{ee}}(\text{vm}_k)) \leq \text{cost}_{\text{bgt}}(\tau_i)$ 
4.     if  $f_{\text{ee}}(\text{vm}_k) \in F(\text{vm}_k)$ 
5.         return  $f_{\text{ee}}(\text{vm}_k)$ 
6.     else
7.         for  $f_k = (\text{vm}_{j, f_{\text{ee}}}, \text{vm}_{j, \text{min}}]$  do
8.             if  $\text{cost}(\tau_i, f_k(\text{vm}_j)) < \text{cost}_{\text{bgt}}(\tau_i)$ 
9.                 return  $f_k$ 
10.            end if
11.        end for
12.    else
13.        for  $f_n = (\text{vm}_{j, f_{\text{ee}}}, \text{vm}_{j, \text{max}}]$  do
14.            if  $\text{cost}(\tau_i, f_n(\text{vm}_j)) < \text{cost}_{\text{bgt}}(\tau_i)$ 
15.                return  $f_n$ 
16.            else
17.                return  $f_{\text{max}}$ 
18.            end if
19.        end for
20. end if
```

4.3 EABL 算法的时间复杂度分析

当系统中采用 $|M|$ 个虚拟机来调度用户提交的含有 $|N|$ 个任务的工作流图时,虚拟机支持的频率平均为 $|V|$ 级,算法 2 的时间复杂度为 $O(|V|/2)$ 。EABL 算法中,步骤 1 和步骤 2 的时间复杂度均为 $O(|M| \times |N|)$;步骤 3 的时间复杂度为 $O(|M| \times |V|)$;消耗时间最多的是步骤 8—步骤 22,此循环体的时间复杂度为 $O(|M| \times |N| \times |V|/2)$ 。因此,不难得出, EABL 算法的时间复杂度为 $O(|M| \times |N| \times |V|)$ 。

4.4 EABL 算法的空间复杂度分析

不难看出,当虚拟机支持 $|V|$ 级频率时,算法 2 的空间复杂度为 $O(1)$ 。EABL 算法中,步骤 1 和步骤 2 均采用递归计算,其空间复杂度均为 $O(|N|)$;步骤 3—步骤 6 均为单次计算,其空间复杂度均为 $O(1)$;步骤 7 采用稳定的快速排序时,空间复杂度为 $O(|N| \log_2 |N|)$;步骤 8—步骤 24 的空间复杂度为 $O(1)$ 。由于 $|N|$ 常为一个大于 2 的正整数,因此 EABL 算法的空间复杂度为 $O(|N| \log_2 |N|)$ 。

4.5 典型的案例

如图 1 所示的工作流图,当采用 3 组虚拟机进行调度时,虚拟机的单位价格如表 3 所列。同时,3 组虚拟机中处理器的 P_{ind} 分别取值为 1.0, 1.1 和 0.6,对应的有效电容 c_{eff} 分别为 0.8, 1.2 和 1.0。根据 EABL 算法中 $URank$ 的降序排列,可得到任务的优先顺序为 $\{\tau_0, \tau_3, \tau_2, \tau_4, \tau_1, \tau_6, \tau_5, \tau_7, \tau_8, \tau_9, \tau_{10}\}$ 。根据前面提供的参数,不难得出 $\text{cost}_{\text{min}}(G) = 528$, $\text{cost}_{\text{max}}(G) = 2387.5$ 。当任务集的总预算成本为 660 时,根据 EABL 算法得到各任务的关键信息,如表 4 所列。

表 3 虚拟机的单位价格
Table 3 Unit price of VMs

vm_k	$p(\text{vm}_k)$
vm_0	4
vm_1	5
vm_2	7

表 4 $\text{cost}(G) = 660$ 时根据 EABL 算法得到的图 1 中各节点的信息

Table 4 Information of each node in Fig. 1 under EABL when $\text{cost}(G) = 660$					
Task	$URank$	$DRank$	cost_{id}	cost_{bgt}	$E(\tau_i)$
τ_0	119.33	0.00	67.59	67.59	24.14
τ_1	87.67	24.00	61.78	71.68	21.43
τ_2	93.00	20.00	55.92	57.11	11.20
τ_3	93.33	26.00	60.59	67.07	11.43
τ_4	90.67	21.00	51.78	59.90	17.62
τ_5	63.33	48.33	76.78	84.55	19.20
τ_6	67.00	52.33	68.59	77.77	24.67
τ_7	51.67	50.67	42.80	43.35	14.10
τ_8	37.00	82.33	61.59	64.94	10.00
τ_9	27.67	76.33	54.97	62.27	19.38
τ_{10}	10.67	108.67	57.59	64.86	19.29

图 1 所示的工作流图在 EABL 算法下的调度效果如图 2 所示。任务 τ_0 的预算成本为 67.59,根据 EABL 算法,最后选择在 vm_1 上以 0.9 倍的频率执行,对应的能量消耗为 24.14。从第二候选任务开始,由于前一个任务调度完将产生一定的松弛成本,因此后面尚未调度的任务将更新其预算成本 cost_{bgt} , 并得到更多的成本开销,以便有更为充裕的成本空间选择最优执行虚拟机,节省更多的能量。由图 2 可以看出,在当前的

预算成本下,EABL算法为各任务在其预算范围内选择了能量高效频率运行任务。任务集最终的实际计算开销 $cost_{act}(G)$ 为 651.39,消耗的总能量 $E_{total}(G)$ 为 192.45。

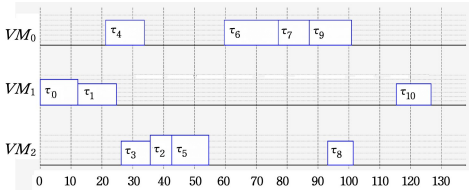


图 2 当 $cost(G)=660$ 时图 1 在 EABL 算法下的调度图
Fig. 2 Scheduling diagram of Fig. 1 under EABL algorithm when $cost(G)=660$

5 实验

实验中采用的评价指标主要为工作流集实际的计算成本 $cost_{act}(G)$ 和任务集运行消耗的总能量 $E_{total}(G)$ 。实验中采用的对比算法为在著名的 HBCS 算法的基础上考虑能量后的算法,即 EA_HBCS 算法和 MECABP 算法。

5.1 实验配置及参数

本文使用 128 个不同计算能力、存储和带宽各异的虚拟机节点,来模拟异构云计算系统中的资源池。云计算模式下的数据中心按需收费,亚马逊 EC2 中虚拟机的收费标准为每小时 0.095 美元至 0.38 美元^[14]。实验中虚拟机间的通信开销为 $1h \leq c_{i,j} \leq 100h$,任务的计算开销为 $1h \leq w_{i,j} \leq 100h$ 。

虚拟机的其他参数为:虚拟机与频率无关的功耗 P_{ind} 服从 $[0.2, 2.0]$ 之间的均匀分布,切换电容参数 C_{eff} 服从 $[0.8, 1.5]$ 之间的均匀分布, α 取值为 3。虚拟机中的各处理器使用归一化的频率,每个处理器支持数量有限且离散的频率等级。

5.2 真实世界工作流图

实验中采用真实世界的两类典型工作流任务集应用作为测试任务集。第一类为快速傅里叶变换(Fast Fourier Transform,FFT)工作流任务集,如图 3(a)所示;第二类为高斯消元工作流任务集,如图 3(b)所示。

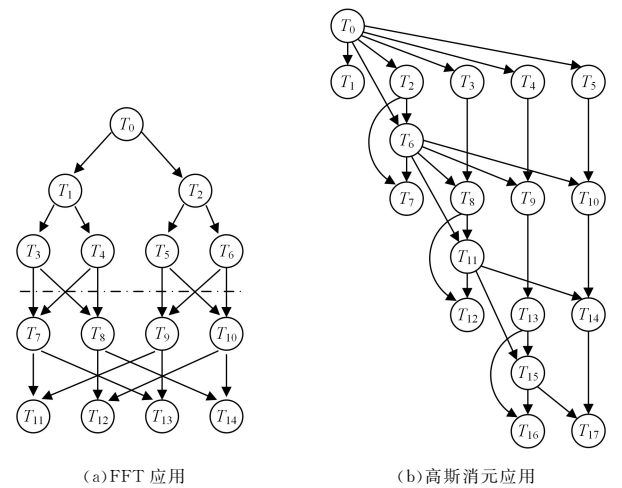


图 3 两种真实世界的工作流应用
Fig. 3 Two real-world workflow applications

5.3 FFT 工作流图

FFT 工作流集在工程、数学、科学计算等领域十分常见,从图 3(a)可以看出,FFT 的并行度很高。FFT 主要包含输入

向量和蝶形计算两部分。此类 DAG 图无出口节点,在实验的过程中可以添加零计算开销的 τ_{exit} ,并添加零通信开销的边与最底层的节点连接。FFT 工作流图的任务节点数随其层数的增加而快速增长,节点数量 $|N|$ 满足关系 $|N| = S_{\delta} = 2^{\delta}$,其中 $\delta > 2$ 且为正整数。

本文实验使用 512 节点的 FFT 工作流图模拟中等规模计算,任务集的最小计算开销 $cost_{min}(G)$ 为 28451,给定的预算成本从 40000 以 5000 为间隔递增到 65000。测试 EABL 算法和对比算法的实验结果如表 5 所列。同时,采用 1024 个任务节点的 FFT 工作流集模拟大规模计算场景,该任务集的最小计算开销 $cost_{min}(G)$ 为 56701,实验中预算成本从 60000 以 10000 为间隔递增到 110000。分别测试 3 种对比算法,实验结果如表 6 所列,不难看出,当给定相同的预算时,本文提出的 EABL 算法可以充分利用预算成本,整个任务集消耗的能量最低。以预算成本为 90000 进行说明,EA_HBCS,MECABP 和 EABL 消耗的能量分别为 31963.37,25268.73 和 20348.37。FFT 是一类并行度很高的工作流集,与 EA_HBCS 和 MECABP 两种算法相比,EABL 在相同的预算成本下,能量消耗平均降低了 31.60% 和 20.09%。

表 5 512 节点的 FFT 工作流图的实际计算开销和能量消耗的对比
Table 5 Comparison of actual computational cost and energy consumption of FFT workflow application with 512 nodes

$ N $	$cost_{bgt}$	EA_HBCS		MECABP		EABL	
		$cost(G)$	$E(G)$	$cost(G)$	$E(G)$	$cost(G)$	$E(G)$
512	40000	39385.13	15719.71	39825.15	13489.83	39960.73	10616.65
512	45000	43127.73	14291.93	44582.3	13125.49	44940.38	10254.3
512	50000	47387.58	14118.82	48923.34	12487.37	49937.49	10025.14
512	55000	53172.37	13731.48	54182.57	12294.53	54797.35	9868.08
512	60000	57138.43	13496.35	56368.52	12283.26	59277.25	9729.06
512	65000	62347.73	13283.03	63138.43	12183.32	64206.66	9677.45

表 6 1024 节点的 FFT 工作流图的实际计算开销和能量消耗的对比
Table 6 Comparison of actual computational cost and energy consumption of FFT workflow application with 1024 nodes

$ N $	$cost_{bgt}$	EA_HBCS		MECABP		EABL	
		$cost(G)$	$E(G)$	$cost(G)$	$E(G)$	$cost(G)$	$E(G)$
1024	60000	59283.18	34183.31	59859.04	29837.91	59995.42	24112.04
1024	70000	69839.35	33178.42	69367.85	28284.63	69978.92	22425.75
1024	80000	78283.82	32174.56	79432.93	26913.18	79960.13	21082.25
1024	90000	88398.73	31963.37	89501.53	25268.73	89955.71	20348.37
1024	100000	98318.22	28146.74	99387.49	24964.32	99946.23	19870.32
1024	110000	107234.49	26928.53	101477.62	24097.51	109701.17	19513.26

5.4 GE 工作流图

GE 工作流图在工程领域被广泛用于解决线性方程组等问题。若用 δ 表示矩阵的维数,那么 GE 工作流图中任务的数量 $|N|$ 满足表达式: $|N| = S_{\delta} = \frac{(\delta^2 + \delta - 2)}{2}$ 。GE 工作流图中部分节点的出度很高,当 GE 的规模较大时,部分节点的出度将进一步增大。

实验中使用 526 节点的 GE 工作流图模拟中等规模的并行任务集,GE 工作流集的 $cost_{min}(G)$ 为 29184,分别测试 EA_HBCS,MECABP 和 EABL 3 种算法。任务集的预算从 40000 以 5000 为间隔递增到 65000,实验结果如表 7 所列。同时,采用 1273 个节点的 GE 工作流图来模拟大规模的计算任务集,分别测试 3 种对比算法。该 GE 工作流集的 $cost_{min}(G)$ 为 70883,测试时使用的预算成本为 75000,80000 至 120000(以

10000 为间隔递增),实验结果如表 6 所列。不难发现,当预算成本相同时,本文提出的 EABL 算法相比 EA_HBCS 和 MECABP 算法更能充分利用预算成本,节省更多的能耗。GE 工作流图的并行度较 FFT 略低;与 EA_HBCS 和 MECABP 两种算法相比,EABL 在同等的预算成本下,能量消耗平均降低了 27.58%和 18.18%。

表 7 526 节点的 GE 工作流图的实际计算开销和能量消耗的对比如

Table 7 Comparison of actual computational overhead and energy consumption of GE workflow application with 526 nodes								
N	cost _{bgt}	EA_HBCS		MECABP		EABL		
		cost(G)	E(G)	cost(G)	E(G)	cost(G)	E(G)	
526	40 000	38 283.12	14 983.52	39 123.83	13 390.23	39 972.29	11 013.38	
526	45 000	43 171.22	14 318.35	44 218.67	13 018.31	44 915.20	10 607.58	
526	50 000	47 832.36	13 956.23	49 125.73	12 451.46	49 944.95	10 350.75	
526	55 000	53 018.57	13 795.86	54 382.64	12 115.45	54 930.86	10 140.93	
526	60 000	56 467.81	13 473.93	58 013.49	11 877.51	59 861.89	9 980.62	
526	65 000	62 832.78	13 345.48	63 794.36	11 798.36	64 854.68	9 924.46	

表 8 1 273 节点的 GE 工作流图的实际计算开销和能量消耗的对比如

Table 8 Comparison of actual computational overhead and energy consumption of GE workflow application with 1273 nodes								
N	cost _{bgt}	EA_HBCS		MECABP		EABL		
		cost(G)	E(G)	cost(G)	E(G)	cost(G)	E(G)	
1273	75 000	74 193.03	41 382.18	74 358.82	36 589.24	74 988.73	30 079.68	
1273	80 000	79 283.25	39 878.21	79 283.01	35 103.82	79 984.29	29 296.01	
1273	90 000	89 183.06	38 237.53	89 415.39	34 274.73	89 968.88	27 556.33	
1273	100 000	99 283.31	36 925.67	99 238.5	33 009.07	99 956.9	26 289.08	
1273	110 000	10 192.37	35 589.17	109 182.23	31 108.39	109 961.5	25 544.46	
1273	120 000	110 844.92	34 189.46	113 72.36	30 181.12	119 942.32	25 074.92	

结束语 本文针对异构云计算系统中预算约束的工作流任务集调度中的节能问题,提出了一种基于预算等级约束的能量感知调度算法 EABL。EABL 算法通过建立高效任务队列,在任务预算开销计算阶段充分利用前一阶段任务调度后剩余的计算成本,来为下一阶段最优执行虚拟机和能量高效执行频率的选择积累相对宽裕的计算成本。最后,通过真实世界典型的大规模 FFT、GE 工作流任务集对 EABL 算法和著名的 EA_HBCS 和 MECABP 进行性能测试。实验结果表明,与 EA_HBCS 和 MECABP 相比,针对 FFT 任务集,EABL 算法的能量消耗分别降低了 31.60%和 20.09%;针对 GE 任务集,EABL 算法的能量消耗分别降低了 27.58%和 18.18%。任务执行过程的高可靠性同样是非常重要的指标,下一步将重点探索预算成本约束下工作任务集的可靠性加强问题。

参 考 文 献

[1] SHARMA N K, REDDY G R M. Multi-objective energy efficient virtual machines allocation at the cloud data center[J]. IEEE Transactions on Services Computing, 2019, 12(1): 158-171.

[2] ZHANG L X, LI K L, XU Y M, et al. Maximizing reliability with energy conservation for parallel task scheduling in a heterogeneous cluster[J]. Information Sciences, 2015, 319: 113-131.

[3] MANSOURI Y, TOOSI A N, BUYYA R. Cost optimization for dynamic replication and migration of data in cloud data centers [J]. IEEE Transactions on Cloud Computing, 2019, 7(3): 705-718.

[4] LI K Q. Scheduling parallel tasks with energy and time constraints on multiple manycore processors in a cloud computing environment[J]. Future Generation Computer Systems, 2018, 82: 591-605.

[5] ZHANG X M, JIA M, GU X M, et al. An energy efficient resource allocation scheme based on cloud-computing in H-CRAN [J]. IEEE Internet of Things Journal, 2019, 6(3): 4968-4976.

[6] ZHANG L X, LI K L, ZHENG W H, et al. Contention-aware reliability efficient scheduling on heterogeneous computing systems[J]. IEEE Transactions on Sustainable Computing, 2018, 3(3): 182-194.

[7] JUAREZ F, EJARQUE J, BADIA R M. Dynamic energy-aware scheduling for parallel task-based application in cloud computing [J]. Future Generation Computer Systems, 2018, 78: 257-271.

[8] ZHANG L X, LI K L, LI C Y, et al. Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems [J]. Information Sciences, 2017, 379: 241-256.

[9] SHOJAFAR M, CORDESCHI N, BACCARELLI E. Energy-efficient adaptive resource management for real-time vehicular cloud services[J]. IEEE Transactions on Cloud Computing, 2019, 7(1): 196-209.

[10] PENG H, WEN W S, TSENG M L, et al. Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment[J]. Applied Soft Computing, 2019, 80: 534-545.

[11] LI Z, GE J, HU H, et al. Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds [J]. IEEE Transactions on Services Computing, 2018, 11(4): 713-726.

[12] ABRISHAMI S, NAGHIBZADEH M, EPEMA D H J. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds[J]. Future Generation Computer Systems, 2013, 29(1): 158-169.

[13] ARABNEJAD H, BARBOSA J G. A budget constrained scheduling algorithm for workflow applications[J]. Journal of Grid Computing, 2014, 12(4): 665-679.

[14] CHEN Y K, XIE G Q, LI R F. Reducing energy consumption with cost budget using available budget preassignment in heterogeneous cloud computing systems[J]. IEEE Access, 2018, 6: 20572-20583.



ZHANG Long-xin, born in 1983, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. His main research interests include cloud computing, scheduling for distributed computing systems and big data analysis.



WEN Hong, born in 1981, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. His main research interests include deep learning and network virtualization.