

异构计算系统中能量感知利润最大化在线算法

张庆辉¹, 李伟东², 张学杰¹

(1. 云南大学 信息学院 云南 昆明 650500; 2. 云南大学 数学与统计学院 云南 昆明 650500)

摘要: 异构计算系统中的任务调度仅以能耗优化为目标, 往往会忽略最大完工时间带来的负面影响。此外, 庞大的机器与任务数量为调度决策带来了极大的时间成本。以异构计算系统管理者单位时间收益最大化为目标, 建立了考虑任务包的能量感知利润最大化问题模型, 并为之设计了一种高效的在线算法。每到达一个用户, 该在线算法能够通过系统当前状态构造多个线性方程组, 并求得利润最大的解, 即当前用户提交任务的分配策略。同时计算了该算法的运行时间复杂度为 $O(nm^4)$ 。通过与另外两种常用算法进行对比, 提出的在线算法能够在多项式时间内, 得到目标值拟最优的调度方案。

关键词: 异构计算系统; 作业调度; 能量感知; 负载均衡; 任务包

中图分类号: TP302

文献标志码: A

文章编号: 1671-6841(2024)01-0047-06

DOI: 10.13705/j.issn.1671-6841.2022250

An Online Algorithm for Energy-aware Profit Maximizing Problem in Heterogeneous Computing System

ZHANG Qinghui¹, LI Weidong², ZHANG Xuejie¹

(1. School of Information Science and Engineering, Yunnan University, Kunming 650500, China;

2. School of Mathematics and Statistics, Yunnan University, Kunming 650500, China)

Abstract: Task scheduling in heterogeneous computing systems only targeted at energy consumption optimization, which could ignore the negative impact of maximum completion time. In addition, the large number of machines and tasks could incur a significant time cost for scheduling decisions. A energy-aware profit maximizing problem considering bag-of-tasks with the goal of maximizing the profit per unit time for managers of heterogeneous computing systems was built, and an efficient online algorithm was designed. For each user, the online algorithm could construct multiple sets of system of linear equations based on the system's current state, and solve the most profitable solution which was the allocation strategy for the tasks submitted by the current user. The running time of the algorithm was $O(nm^4)$. The proposed algorithm was compared with the other two common algorithms. The proposed online algorithm could obtain a quasi-optimal scheduling scheme in polynomial time.

Key words: heterogeneous computing system; task scheduling; energy-aware; load balancing; bag-of-tasks

0 引言

随着数据中心能耗的快速增长, 异构计算系统

中关注于能效的任务调度越来越重要。最近学界提出了一种名为任务包 (bag-of-tasks) 的静态调度模型^[1]。与以往的经典调度模型相比, 某台机器上的固定执行时间 (estimated time to compute, ETC) 取决

收稿日期: 2022-08-20

基金项目: 国家自然科学基金项目 (12071417, 61762091, 62062065); 云南大学第十三届研究生科研创新项目 (2021Z079)。

第一作者: 张庆辉 (1998—), 男, 硕士研究生, 主要从事边缘计算中资源分配问题研究, E-mail: zhangqinghui@mail.ynu.edu.cn。

通信作者: 张学杰 (1965—), 男, 教授, 主要从事高性能计算以及云计算研究, E-mail: xjzhang@ynu.edu.cn。

于任务类型与机器类型。通过这一概念,能把异构计算系统中心成千上万的任务进行分类,而分类后的类型数量相对较少。如果单独地为每一个任务进行决策,那将耗费难以接受的时间。而任务包的调度模型很好地限制了问题规模,这也使得为该问题设计一种能得到拟最优调度的高效算法成为可能^[2-3]。

经典的能量感知调度模型旨在最小化任务包所消耗的能量或最大完工时间。然而,对于异构计算系统中心运营商而言,将每个单位时间的运行利润最大化会带来更多的经济收益,其中利润等于用户为一个任务包支付的费用减去执行该任务包消耗的电力成本。通过综合考虑能源成本和最大完工时间,实现单位时间利润最大化的目标,即考虑任务包的能量感知利润最大化(energy-aware profit maximizing, EAPM)问题。TarpLee 等以此为目标提出了一种新的异构计算系统调度模型,该模型具有机器类型数与任务数都十分有限的特征^[3]。通过使用一种新颖的基于线性规划(linear programming, LP)的舍入算法,设计了一个能够得到接近最优调度的高效算法。

TarpLee 等用一个最大完工时间下界代替了原本的最大完工时间^[3]。最大完工时间下界是将任务平均分配给所有机器时的完工时间。这个下界与真实的最大完工时间是有一定差距的。因此,该数学模型是不准确的,而且在基于 LP 的舍入步骤过程中,能耗成本可能会增加。这导致即便可以通过使用基于匹配的舍入技术来改善,但执行时间会随着问题规模的扩大而急剧上升,使该算法在大型数据中心无法有很好的表现。

本文的主要贡献如下:1) 为 EAPM 问题建立了一个准确的数学模型,该模型能精确计算每到达一个用户并分配其任务包后系统的最大完工时间;2) 提出了一个时间复杂度为 $O(nm^4)$ 的在线算法,该算法在每个用户到达时构造多个线性方程组,这些线性方程组中最好的结果,便是当前针对该用户任务包的调度结果;3) 通过对比实验,说明了本文提出算法的优越性。

1 相关工作

最近几十年有大量关于异构计算系统中任务调度模型的研究。Braun 等比较了 11 种静态启发式算法,他们将一类互相独立的任务映射到异构分布式计算系统,来最小化最大完工时间^[4]。Dai 等在

包含两台平行机的系统中,设计了一种半在线算法,能很好地限制机器的最大完工时间^[5]。针对考虑任务包的调度模型,TarpLee 等提出了一种基于线性规划的资源分配算法,能高效地给出最小化最大完工时间的调度方案^[2]。胡逸骢针对面向异构计算集群的任务调度和能量消耗问题,提出一种面向异构计算系统的能量感知任务调度算法^[6]。

Friese 等针对任务包这种情形提出了一种改进的多目标遗传算法来生成多个不同的调度方案,能很好地平衡能源消耗和最大完工时间之间的得失^[1,7]。他们还创建了一个工具,该工具能帮助系统管理员对系统性能和系统能量分配进行权衡^[8]。Zhang 等设计了一个整数线性双目标优化模型,并提出了两阶段的启发式分配算法以找到高质量的可行解决方案^[9]。除此之外,追求能量感知利润最大化的目标也能很好地平衡最大完工时间和能耗。Li 等针对考虑任务包的能量感知利润最大化问题,设计了一个最坏情况即近似比接近 2 的近似算法^[10]。随后又提出了一个针对该问题的多项式时间近似算法,该算法同样能在某些情况下有接近 2 的近似比效果^[11]。

在云计算环境中,云资源管理同样是云供应商的一个重要内容。Khemka 等为能源受限的环境设计了四种能量感知的资源分配启发式方法,目的是使系统获得的总效用最大化^[12]。姜春茂等提出面向实时云任务的细粒度任务合并调度算法,在满足用户 SLA 的前提下,能够有效降低云能耗^[13]。Zhang 等通过拍卖机制对云计算虚拟资源进行分配和定价,以提升资源提供商的社会福利^[14]。

2 在线调度模型

在一个异构计算机系统中包含了 m 种不同的机器类型和 n 种不同类型的用户。用户 i 提交的任任务包中的任务相互独立,数量为 a_i ^[4],执行这类任务能产生的收益为 p_i 。 $ETC = (ETC_{ij})$ 是一个 $n \times m$ 维矩阵, ETC_{ij} 是用户 i 的任务在机器 j 上执行所需的固定执行时间; $APC = (APC_{ij})$ 同样是一个 $n \times m$ 维矩阵,其中 APC_{ij} 是用户 i 的任务在机器 j 上执行所需要的平均功率消耗(average power consumption, APC)^[3]。 x_{ij} 表示用户 i 的任务分配给机器 j 执行的任务数。对于一个可行解 $\mathbf{x} = (x_{ij})$,机器 j 的负载可以定义为

$$L_j = \sum_{i=1}^n ETC_{ij}x_{ij} \quad (1)$$

所有机器的最大完工时间 $MS(\mathbf{x})$ 为

$$MS(\mathbf{x}) = \max_j L_j. \quad (2)$$

相应的, n 个用户的能量消耗为

$$E(\mathbf{x}) = \sum_{j=1}^m \sum_{i=1}^n x_{ij} APC_{ij} ETC_{ij}. \quad (3)$$

用 c 表示每个单位能耗的成本, EAPM 问题可以用非线性整数规划表示:

$$\max \frac{\sum_{i=1}^n p_i - cE(\mathbf{x})}{MS(\mathbf{x})}, \quad (4)$$

$$\text{s. t. } \sum_{j=1}^m x_{ij} = a_i, \forall i = 1, 2, \dots, n. \quad (5)$$

$$\sum_{i=1}^n x_{ij} ETC_{ij} \leq MS(\mathbf{x}), \forall j = 1, 2, \dots, m, \quad (6)$$

$$x_{ij} \in \mathbf{N}, \forall i, j. \quad (7)$$

目标函数(4)要最大化单位时间的收益, \mathbf{x} 是决策向量。约束(5)确保了每一个用户的每一个任务都被分配给某个机器。由于最大化单位时间收益的目标等价于最小化最大完工时间, 约束(6)确保了 $MS(\mathbf{x})$ 是所有机器的最大完工时间。

然而在实际场景中, 当某个用户到达时就需要在机器不知道未到达用户的信息的情况下分配该用户的所有任务。因此, 研究考虑任务包的 EAPM 问题的在线算法是很有必要的。该在线算法考虑用户 i 的任务会在用户 $i+1$ 到达之前就被分配, $i = 1, 2, \dots, n-1$ 。但是, 当用户 i 提交的任务数非常大时, 不能逐个分配这些任务。因此, 为考虑任务包的 EAPM 问题设计一个高效的在线算法是很有必要的。

3 在线算法

对于每一个用户 i , 用 L_j^i 和 E^i 表示分配结束前 i 个用户的任务之后机器 j 的负载和系统的总能耗。初始情况下 $L_j^0 = 0, j = 1, 2, \dots, m, E^0 = 0$ 。根据定义, 对于任意 $i = 1, 2, \dots, n$,

$$L_j^i = \sum_{k=1}^i x_{kj} ETC_{kj}, \quad (8)$$

$$E^i = \sum_{k=1}^i \sum_{j=1}^m x_{kj} APC_{kj} ETC_{kj}. \quad (9)$$

当用户 i 到达时, 确定 x_{ij} 的值并让 $\sum_{j=1}^m x_{ij} = a_i$, 此时的目标值为

$$\frac{\sum_{k=1}^i p_k - cE^{i-1} - c \sum_{j=1}^m x_{ij} APC_{ij} ETC_{ij}}{MS^i}, \quad (10)$$

其中:

$$MS^i = \max_j L_j^i, L_j^i = L_j^{i-1} + x_{ij} ETC_{ij}, \forall i, j.$$

令 $C_i = c \sum_{j=1}^m x_{ij} APC_{ij} ETC_{ij}$, 那么该问题可以形式化地表示为

$$\max \left(\sum_{k=1}^i p_k - cE^{i-1} - C_i \right) / MS^i, \quad (11)$$

$$\text{s. t. } \sum_{j=1}^m x_{ij} = a_i, \quad (12)$$

$$L_j^{i-1} + x_{ij} ETC_{ij} \leq MS^i, x_{ij} \in \mathbf{N}, j = 1, 2, \dots, m. \quad (13)$$

约束(13)等价于 $x_{ij} \leq \lfloor \frac{MS^i - L_j^{i-1}}{ETC_{ij}} \rfloor$ 。

为了便于操作, 将服务于用户 i 的任务的机器按照 $APC_{ij} ETC_{ij}$ 降序排序。不失一般性, 假设

$$APC_{i1} ETC_{i1} \geq APC_{i2} ETC_{i2} \geq \dots \geq APC_{im} ETC_{im}, \quad (14)$$

我们的算法是基于引理 1 实现的。

引理 1 存在一个最优解, 该最优解符合

$$x_{i1} = \dots = x_{i(\tau-1)} = 0, x_{ij} = \lfloor \frac{MS^i - L_j^{i-1}}{ETC_{ij}} \rfloor, j = \tau, \dots, m,$$

其中: $\tau \in \{1, 2, \dots, m\}$ 。

证明 假设已知整数规划(11)最优解中 MS^i 的值, 那么规划(11)的目标函数便等价于最小化

$\sum_{j=1}^m x_{ij} APC_{ij} ETC_{ij}$ 的值, 其中的 L_j^{i-1} 和 ETC_{ij} 是两个常量。很明显, 在约束(12)和(13)限制下, 为了使

$\sum_{j=1}^m x_{ij} APC_{ij} ETC_{ij}$ 的值尽可能小: 若 $APC_{ij} ETC_{ij}$ 值小, 则使其对应的 x_{ij} 大; 若 $APC_{ij} ETC_{ij}$ 值大, 则使其对应的

x_{ij} 小。

对于规划(11)的一个最优解 $(x_{i1}, x_{i2}, \dots, x_{im})$, 假设 τ_1 是使 $x_{i\tau_1} > 0$ 成立的最小机器下标。如果存在一台机器 $\tau_2 (\geq \tau_1)$, 有 $x_{i\tau_2} < \lfloor (MS^i - L_{\tau_2}^{i-1}) / ETC_{i\tau_2} \rfloor$, 令

$$x'_{ij} = \begin{cases} x_{ij} - 1, & j = \tau_1, \\ x_{ij} + 1, & j = \tau_2, \\ x_{ij}, & j \neq \tau_1, \tau_2. \end{cases} \quad (15)$$

很容易证明 $(x'_{i1}, x'_{i2}, \dots, x'_{im})$ 是规划(11)的一个可行解。由于式(14)重排序使得 $APC_{i\tau_1} ETC_{i\tau_1} \geq APC_{i\tau_2} ETC_{i\tau_2}$, 该可行解的目标值会比 $(x_{i1}, x_{i2}, \dots, x_{im})$ 的目标值更小。对所有机器 $j \geq \tau$ (τ 是使 $x_{i\tau} > 0$ 的最小机器下标), 重复上面的过程, 直到 $x_{ij} = \lfloor (MS^i - L_j^{i-1}) / ETC_{ij} \rfloor$ 。这预示着能找到一个最优解满足

$$x_{i1} = \dots = x_{i(\tau-1)} = 0,$$

$$x_{ij} = \lfloor \frac{MS^i - L_j^{i-1}}{ETC_{ij}} \rfloor, j = \tau + 1, 2, \dots, m.$$

因此,该引理成立。

对于每个 $\tau = 1, 2, \dots, m$, 只需要考虑部分的决策变量 $x_{ij} (j = \tau, \dots, m)$ 。此时想要得到用户 i 到达时的所有 x_{ij} 值以及 MS^i 的值,通过计算可得预期结果:

$$\begin{cases} \sum_{j=\tau}^m x_{ij} = a_i, \\ x_{ij} = \frac{MS^i - L_j^{i-1}}{ETC_{ij}}, j = \tau, \dots, m. \end{cases} \quad (16)$$

式(16)共有 $m - \tau + 2$ 个等式和 $m - \tau + 2$ 个未知数,未知数为 MS^i 和 $x_{ij} (j = \tau, \dots, m)$ 。因此,在多项式时间内对式(16)进行求解。对于每一个 $\tau = 1, 2, \dots, m$, 得到一组 x_{ij} 的值。比较这 m 组可行解的目标值,便能得到当前的最优调度。将 $\lceil x_{ij} \rceil$ 个用户 i 的任务分配给机器 j , 直到所有任务都被分配。

算法1 在线算法

输入: m, n, ETC, APC 以及用户到达序列。

输出: x 。

1) for $i = 1, 2, \dots, n$

2) 重索引下标使得 $APC_{i1}ETC_{i1} \geq \dots \geq$

$APC_{im}ETC_{im}$

3) for $\tau = 1, 2, \dots, m$

4) 解方程组(16)得 x_{ij}^τ

5) 比较这 m 个解,找到使得公式(11)的值最大的 x_{ij}

6) for $j = 1, 2, \dots, m$

7) 将 $\lceil x_{ij} \rceil$ 个用户 i 的任务分配给机器 j , 直到所有任务都被分配

8) 更新机器 j 对应的 L_j

接下来计算在线算法的时间复杂度。该算法中最主要耗费时间的是步骤4)中的解线性方程组。通过矩阵运算来对方程组进行求解,花费的时间为 $(m - \tau + 2)^3$ 。对于每个到达用户,需要求解 m 个线性方程组,共有 n 个用户到达。因此,此算法的时间复杂度为 $O(nm^4)$ 。

4 实验评估

4.1 实验环境

为了进行对比,除了本文提出的在线方法(Online),我们还使用了贪心算法(Greedy)和平均分配算法(Average)来解决EAPM问题。Greedy算法将

会在一个用户到达时,将该用户的整个任务包分配给能使 $ETC \cdot APC$ 值最小的那台机器,而Average将会把该用户的整包任务平均分配给每台机器。使用C++编程语言实现上述算法,并在以下硬件配置环境中进行了实验:CPU为Intel i7-10700,8核16线程2.9 GHz,16 GB内存及1 TB硬盘。实验的部分数据源于TarpLee在OpenBenchmark基础上进行了扩充的数据集,共有包含9种机器类型与30种任务类型^[15]。

所有的实验中令 $c = 1$ 。若不考虑最大完工时间,将 E_{\min}^i 定义为用户 i 的能耗下界, $E_{\min}^i = \min_j ETC_{ij}APC_{ij}$ 。此外, $p_i = \gamma E_{\min}^i$, 那么 $\gamma = p_i / E_{\min}^i$ 便是一个能够影响用户 i 出价的参数。在实际工业活动中,高性能中心的管理者为了盈利,一般不会选择较小的 $\gamma (\gamma \leq 1)$ 。根据TarpLee得出的结论,当 γ 足够大时 ($\gamma > 1.5$), 寻求单位时间效益最大化时分配结果会更趋向于最小化最大完工时间^[15]。因此,在本节的实验中,仅考虑 $\gamma \in (1, 1.5]$ 。

4.2 不同 γ 值对系统的影响

本实验是在上述9种机器类型和30种任务类型的基准上进行的。考虑30个用户按任意顺序提交类型各不相同的任务包,用户 i 的任务包中的任务数 $a_i \in [200, 1000]$ 。随着 γ 的变化,三种方法得到的目标值(式(4))变化如表1所示。为了减小随机性带来的影响,每次 γ 取值后重复100次实验并将结果取平均值,结果见表1。从表1可以看到本文提出的Online方法在所有情况下都优于另外两种方法。当 γ 较小时,Online方法与Greedy方法得到的目标值差距并不大,但当 γ 逐渐增大后Online方法便体现了其优越性。Average方法与另外两种方法得到的目标值有着较大差距。它虽然能使得系统的最大完工时间最小,但是由于并未考虑成本的原因,会使得最终的目标值变得很差。

4.3 不同用户数对系统的影响

本实验研究了用户数 (n) 从30逐渐增长到2000时对系统最终目标值的影响。为了更全面地评估 n 的增长带来的影响,取 $\gamma = 1.2, 1.3$ 和 1.5 进行对比。用户 i 提交的任务包中的任务数 $a_i \in [200, 1000]$ 。

从图1中可以看到当用户数较小时,Online方法得到的目标值更大,当用户数不断增加,Online方法与Greedy方法得到的目标值几乎变得相等。这是由于当 n 变得足够大时,总的任务数也会增加,这会使得两种方法给出的调度方案产生的最大完工时间逐渐靠近最优调度所产生的最大完工时间。

表 1 不同 γ 值对目标值的影响

Table 1 The object value was effected by different γ values

γ	Online	Greedy	Average
1.05	22.23	22.63	-382.80
1.10	45.26	43.44	-347.77
1.15	70.47	67.89	-312.73
1.20	99.50	90.52	-277.70
1.25	130.69	113.15	-242.67
1.30	162.65	135.78	-207.63
1.35	190.94	158.41	-172.60
1.40	224.40	181.04	-137.56
1.45	255.72	203.68	-102.53
1.50	297.67	226.31	-67.50

从图 2 可以看出三种方法的执行时间都在随着用户数的增加而增加。虽然 Online 方法的执行时间最长,但它得到的目标值也是最大的,即使在 $n = 2\ 000$ 时其执行时间也是 ms 级的。

4.4 用户到达顺序对系统的影响

在实际场景中,用户到达后所提交的任务数是无法预知的。为了评估用户到达顺序对系统的影响,令用户数 $n = 30$, 类型为随机。如果用户 i 提交的包中的任务数 $a_i > 500$, 则为大任务包;若用

户 i 提交的包中的任务数 $a_i < 200$, 则为小任务包。对比四种到达顺序对应的目标值,“BaS”为前一半的用户提交大任务包,后一半用户提交小任务包;“SaB”为前一半的用户提交小任务包,后一半用户提交大任务包;“Rand”为提交大任务包与小任务包的用户随机到达;“Equal”为所有用户提交的包任务数相同,为 400。

从图 3 可以看到,在所有情况下 Online 方法都能得到最好的目标值结果。此外,用户到达的顺序并未对目标值造成明显的影响,只有当 SaB 情形时 Online 方法与 Greedy 得到的目标值会稍微降低。这是由于先将小任务包分配之后,为了不引起最大完工时间的快速增长,大包任务到达时有较小可能被分配给成本更大的机器进行执行。

5 总结

我们提出的 Online 的方法,通过新颖的方法构造线性方程组得到最终的分配结果,其执行时间依赖用户数以及机器种类数,并通过实验证明此方法能得到令人满意的结果。

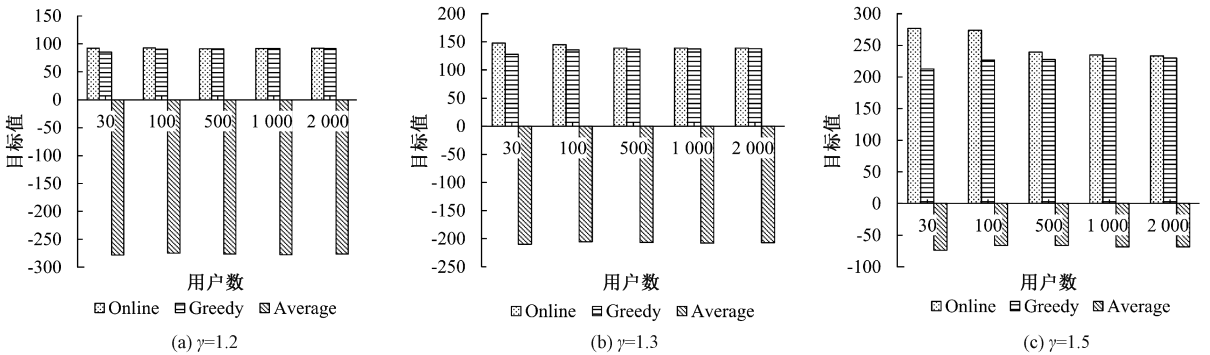


图 1 三种方法的目标值随着 n 增长的变化

Figure 1 Objective values of three methods with varying n

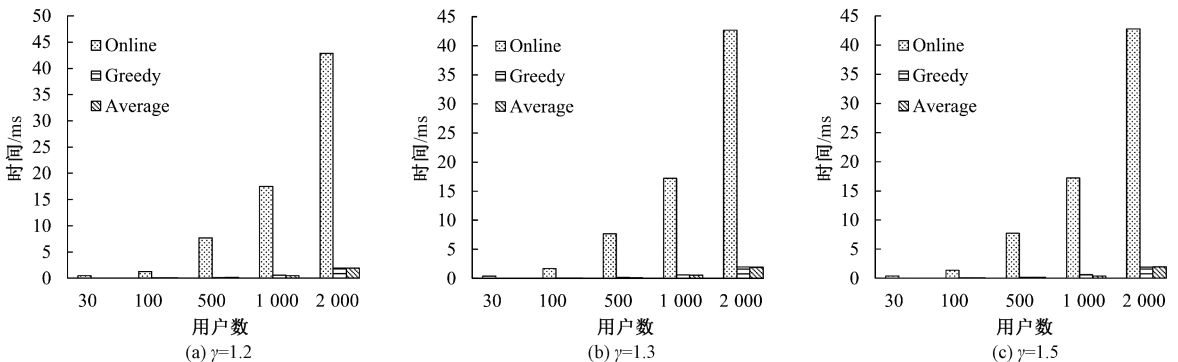


图 2 三种方法的执行时间随着 n 增长的变化

Figure 2 Execution time of three methods with varying n

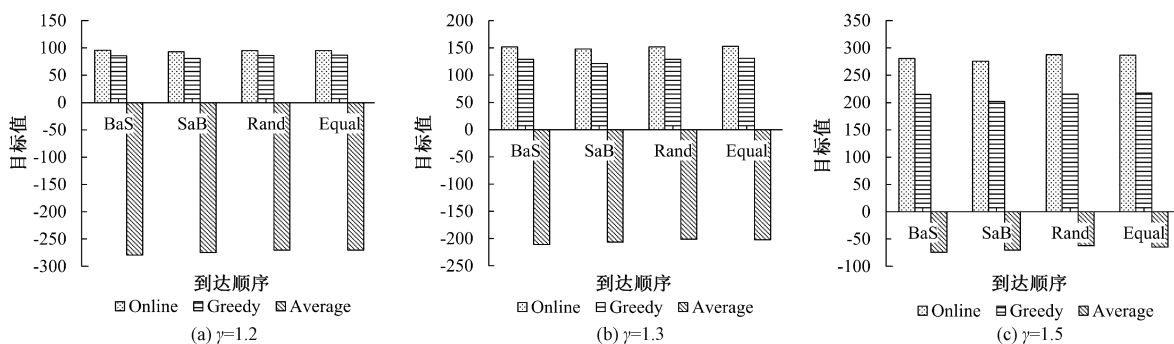


图3 不同用户到达顺序对目标值的影响

Figure 3 The impact of different user arrival sequences on the objective value

参考文献:

- [1] FRIESE R D, BRINKS T, OLIVER C, et al. Analyzing the trade-offs between minimizing makespan and minimizing energy consumption in a heterogeneous resource allocation problem[C]//The Second International Conference on Advanced Communications and Computation. Piscataway: IEEE Press, 2012: 81–89.
- [2] TARPLEE K M, FRIESE R, MACIEJEWSKI A A, et al. Scalable linear programming based resource allocation for makespan minimization in heterogeneous computing systems[J]. Journal of parallel and distributed computing, 2015, 84: 76–86.
- [3] TARPLEE K M, FRIESE R, MACIEJEWSKI A A, et al. Energy and makespan tradeoffs in heterogeneous computing systems using efficient linear programming techniques[J]. IEEE transactions on parallel and distributed systems, 2016, 27(6): 1633–1646.
- [4] BRAUN T D, SIEGEL H J, BECK N, et al. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems[J]. Journal of parallel and distributed computing, 2001, 61(6): 810–837.
- [5] DAI B F, LI J P, LI W D. Semi-online hierarchical scheduling for bag-of-tasks on two machines[C]//Proceedings of the International Conference on Computer Science and Artificial Intelligence. New York: ACM Press, 2018: 609–614.
- [6] 胡逸骢. 面向高性能计算的能耗感知任务调度算法及应用[D]. 长沙: 湖南大学, 2019.
HU Y K. Energy-aware task scheduling algorithms and application for high-performance computing[D]. Changsha: Hunan University, 2019.
- [7] FRIESE R D, KHEMKA B, MACIEJEWSKI A A, et al. An analysis framework for investigating the trade-offs between system performance and energy consumption in a heterogeneous computing environment[C]//IEEE International Symposium on Parallel & Distributed Processing, Workshops and PHD Forum. Piscataway: IEEE Press, 2013: 19–30.
- [8] FRIESE R D, BRINKS T, OLIVER C, et al. A machine-by-machine analysis of a bi-objective resource allocation problem[C]//The 9th International Conference on Parallel and Distributed Processing Techniques and Applications. Las Vegas: CSREA Press, 2013: 3–9.
- [9] ZHANG X L, LIU X, LI W D, et al. Trade-off between energy consumption and makespan in the mapreduce resource allocation problem[C]//International Conference on Artificial Intelligence and Security. Cham: Springer International Publishing, 2019: 239–250.
- [10] LI W D, LIU X, ZHANG X J, et al. A task-type-based algorithm for the energy-aware profit maximizing scheduling problem in heterogeneous computing systems[C]//The 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Piscataway: IEEE Press, 2015: 1107–1110.
- [11] LI W D, LIU X, CAI X B, et al. Approximation algorithm for the energy-aware profit maximizing problem in heterogeneous computing systems[J]. Journal of parallel and distributed computing, 2019, 124: 70–77.
- [12] KHEMKA B, FRIESE R, PASRICHA S, et al. Utility maximizing dynamic resource management in an oversubscribed energy-constrained heterogeneous computing system[J]. Sustainable computing: informatics and systems, 2015, 5: 14–30.
- [13] 姜春茂, 王凯旋. 基于三支队列的实时云任务节能调度算法[J]. 郑州大学学报(理学版), 2019, 51(2): 66–71.
JIANG C M, WANG K X. Real-time cloud tasks schedule algorithm for saving energy based on tri-queue system[J]. Journal of Zhengzhou university (natural science edition), 2019, 51(2): 66–71.
- [14] ZHANG J X, XIE N, ZHANG X J, et al. An online auction mechanism for cloud computing resource allocation and pricing based on user evaluation and cost[J]. Future generation computer systems, 2018, 89: 286–299.
- [15] TARPLEE K M, MACIEJEWSKI A A, SIEGEL H J. Energy-aware profit maximizing scheduling algorithm for heterogeneous computing systems[C]//The 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Piscataway: IEEE Press, 2014: 595–603.