

DOI: 10.7652/xjtuxb201602001

面向能耗优化的云渲染系统任务调度策略

李谦, 伍卫国, 曹莹方, 王蕾
(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 针对云渲染系统中由于渲染节点与任务不匹配调度而带来的能耗浪费问题,提出一种通过任务调度方式来优化系统能耗的策略。为了形式化描述系统的整体能耗,综合考虑节点空闲能耗和任务运行能耗,建立渲染任务能耗模型;以降低系统总体能耗为优化目标,根据渲染任务之间无依赖性的特点,将任务调度序列拆分成子序列,利用模拟退火思想,通过优化子序列任务调度提高节点利用率、减少节点空闲能耗,以此降低系统全局任务的能耗;采用矩阵存储子序列任务的能耗,以空间换时间的方式降低策略的时间复杂度。实验结果表明:该策略在多渲染作业环境中能耗优化效果比先进先出算法提升了 43.4%,比能耗感知的调度算法提升了 6.7%,能够有效降低云渲染系统的总体能耗,同时具有良好的扩展性,使云渲染系统的能耗效率和整体性能得到提升。

关键词: 能耗模型;任务调度;云渲染

中图分类号: TP399 **文献标志码:** A **文章编号:** 0253-987X(2016)02-0001-06

A Task Scheduling Strategy With Energy Optimization for Cloud Rendering Systems

LI Qian, WU Weiguo, CAO Yingfang, WANG Lei
(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A task scheduling strategy with optimized energy consumption for cloud rendering systems is proposed to solve the problem that the mismatching task scheduling on render nodes causes a great waste of energy consumption. A rendering task energy consumption model is presented to describe formally the overall energy consumption of a system and takes both the idle and the task running energy consumptions of each node into account. The optimization object is to reduce the overall energy consumption of the system, and the strategy divides the task scheduling sequence into subsequences based on the non-dependence characteristic among rendering tasks. The simulated annealing ideology is used to optimize the scheduling of the subsequence tasks, to improve the utilization ratio of the nodes and to reduce the idle energy consumption of nodes so that the energy consumption for the overall system is reduced. Moreover, the strategy adopts a way of space in time to reduce the time complexity by using a matrix to store the energy of subsequence tasks. Experimental results and comparisons with the FIFO algorithm and EMRSA (energy-aware MapReduce scheduling) algorithm in a multi jobs measurement show that the energy optimization performance of the proposed strategy has improved about 43.4% and 6.7%, respectively, that is, the proposed strategy effectively reduces

the overall energy consumption for cloud rendering systems. Moreover, the proposed strategy possesses better expansibility. It can be concluded that the proposed strategy can improve the energy efficiency and overall performance of cloud rendering systems.

Keywords: energy consumption model; task scheduling; cloud rendering

近年来动漫产业发展迅猛,3D技术被广泛应用到电视、电影制作等领域。作为动漫制作的核心环节之一,渲染决定了动漫作品的质量和视觉效果^[1]。由于需要求解复杂的光照模型,渲染时间成为影视制作的瓶颈。利用云计算的弹性服务和可扩展性等特性^[2],为按时完成渲染任务,许多影视制作公司开始使用云渲染模式,它利用渲染应用的帧与帧之间的天然并行性特征,通过划分任务在高性能计算机系统上并行渲染,达到缩短时间的目的。云渲染将抽象的渲染资源以一种服务的形式提供给用户,用户只需配置相应的渲染参数,渲染计算则完全交给云渲染平台。

随着云渲染模式的广泛应用,许多计算资源和数据资源集中到云端,给系统能耗的高效管理带来了更大的挑战^[3],如何降低云渲染能耗受到越来越广泛的关注。渲染系统中节能调度问题是节省渲染系统中的任务分配处理器等资源,使得所有渲染任务较早完成,同时能量消耗尽量少^[4]。与传统的任务调度相比,节能调度的主要目标不是最小化任务的总体完成时间,而是尽量减少使用渲染资源和占用时间,提高资源利用率,在系统性能与能耗之间寻求平衡^[5-6]。Maheshwari等通过动态改变系统配置参数,根据集群中任务的负载情况调整节点个数^[7]。Pietri等在系统负载平衡状态下建立CPU使用率模型,使用反馈理论调整计算节点的CPU主频,在维持系统负载平衡的同时减少系统能耗^[8]。

基于运行任务的特征减少节点的计算能耗也是有效的节能方式,例如采用处理器动态电压频率调整等底层节能技术^[9];采用虚拟化技术,将多个虚拟机整合到同一个物理机上,关闭或挂起空闲的节点来达到节能的目的^[10]。另外,基于任务负载的调度策略也可以实现能耗的优化。根据计算节点性能及任务特征,Mashayekhy等将能耗因素结合任务调度机制,提出了能耗感知的调度模型^[11]。该模型在服务水平协议的约束下,为每个任务分配最佳执行节点来实现最小化任务功耗的目标。但是对于任务整体而言,当前任务占用的节点有可能也是其他任务能耗最优的运行节点,当其他任务分配到此节点上产生的能耗能够降低任务整体的功耗时,这种基

于局部能耗最优的策略便不利于整体能耗的最优配置。Guzek等将分布式系统中任务调度问题转化成多目标约束条件下的性能优化问题^[12]。

通常,渲染系统能耗主要由任务运行能耗和节点空闲能耗两部分组成。当前能耗优化的研究分别从空闲节点能耗优化和系统任务运行能耗优化两方面改善系统能耗问题,但它们缺少对二者的整体考虑。针对上述问题,本文结合渲染应用特点和作业调度算法建立了渲染任务能耗模型;基于该模型提出了面向能耗优化的任务调度算法,根据模拟退火思想,该算法将任务调度序列拆分成为子序列,通过优化子序列的能耗来实现全局任务的能耗优化。

1 能耗模型

1.1 任务形式化描述

首先,对任务调度过程涉及的等待任务列表和空闲资源列表状态进行描述。在任意时刻 t ,系统等待作业列表为 $J_0, J_1, \dots, J_k$,空闲资源列表为 $R_0, R_1, \dots, R_n$,由 K 个作业的所有任务组成的任务列表为 $l_0, l_1, \dots, l_m$ 。其中,空闲资源个数为 n ,任务个数为 m ,假设 n 和 m 满足关系 $m \geq n$ 。由于渲染应用的最小单位是帧,且帧与帧之间相互独立,不存在依赖关系,因此每一个任务对应一帧,所有任务列表也可以表示为 $f_0, f_1, \dots, f_m$,这里统一将一帧称为一个任务。云渲染系统通常采用三级调度策略,即首先选中一个调度队列;再选中该队列中合适的作业;最后选择作业中合适的任务将其加入任务调度列表分配给相应的渲染节点。由于前两级调度并不会影响任务的能耗,因此直接将建模分析的目标集中在任务调度。经过调度之后,每个渲染节点都维护一个运行任务列表,假设所有节点 $L_0, L_1, \dots, L_n$ 当前任务状态为

$$l_0, l_1, \dots, l_{r-1}, l_r, l_{r+1}, \dots, l_{2r-1}, \dots, l_{nr}, l_{nr+1}, \dots, l_m \quad (1)$$

1.2 任务能耗建模

经过调度之后,节点运行任务列表中的任务与等待任务列表并不相同。在渲染任务提交运行之后,各个渲染节点 $T_0, T_1, \dots, T_n$ 完成所有任务的时间列表为

$$t_0, t_1, \dots, t_{r-1}, t_r, t_{r+1}, \dots, t_{2r-1}, \dots, t_{nr}, t_{nr+1}, \dots, t_m \quad (2)$$

以节点 i 为例对单个节点的能耗进行描述。 T_i 表示节点 i 运行任务的总时间

$$T_i = \sum_{j=ir}^{(i+1)r-1} t_j \quad (3)$$

根据式(3)计算出所有节点的运行时间,式(4)描述了 $T_{\max}$ 的计算过程

$$T_{\max} = \max(T_0, T_1, \dots, T_n) \quad (4)$$

对于第 i 个节点的运行能耗 E_i ,式(5)描述了其计算过程

$$E_i = W_i + I_i \quad (5)$$

式中: W_i 表示该节点执行完任务列表 $L[l_{ir}, l_{ir+1}, \dots, l_{(i+1)r-1}]$ 中所有任务的能耗,该能耗由节点 i 的功率 P_i 和任务运行时间决定; I_i 表示该节点在本次调度中处于空闲状态下所需的能耗,该能耗由节点 i 的空闲功率 p_i 和空闲时间决定。假定当且仅当节点运行完对应列表中的所有任务后才会处于空闲状态。式(6)和式(7)分别描述了上述两种能耗的计算过程

$$W_i = \sum_{j=ir}^{(i+1)r-1} P_i t_j \quad (6)$$

$$I_i = p_i (T_{\max} - T_i) \quad (7)$$

将式(6)和式(7)带入式(5)可以得到第 i 个节点的任务能耗

$$E_i = \sum_{j=0}^{(i+1)r-1} P_i t_j + p_i (T_{\max} - T_i) \quad (8)$$

在节点 i 的能耗计算模型基础上,对于集群节点列表 $R_0, R_1, \dots, R_n$,集群能耗的计算过程为

$$E = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} P_i t_j + \sum_{i=0}^{n-1} p_i (T_{\max} - T_i) \quad (9)$$

集群能耗包括集群中任务运行时能耗和节点空闲能耗。将式(3)带入式(9),经过整理得到集群节点的最终能耗计算模型为

$$E = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (P_i - p_i) t_j + \sum_{i=0}^{n-1} P_i T_{\max} \quad (10)$$

由式(10)可知,渲染任务经过调度运行之后的能耗由两部分组成。第一部分与单个节点的功率、空闲功率和单个任务在渲染节点上的运行时间有关,假设单个节点的功率和空闲功率为常量。第二部分与单个节点的空闲功率和所有节点的任务运行时间有关。同一个节点处理所有任务的时间与任务的个数密切相关,任务调度策略通过影响任务运行的目标节点,从而影响任务的能耗。

2 基于模拟退火的能耗优化调度策略

2.1 任务调度的初始化

模拟退火算法是局部最优化动态规划算法的扩展^[13]。对于面向能耗优化的任务调度问题,以局部能耗最优为目标将每个任务的调度过程描述成为一个相似的子问题,首先根据模拟退火算法的思想完成任务调度的初始设置。

(1)调度问题的解空间。对于等待任务队列 $l_0, l_1, \dots, l_{m-1}$ 和资源队列 $R_0, R_1, \dots, R_{n-1}$,调度问题的解空间就是 m 个任务和 n 个节点形成的映射关系调度序列,该序列描述为 $S: l_{0i}, \dots, l_{(m-1)j}$ 。其中, S 为任务调度队列; l_{ij} 表示将第 i 个任务调度到第 j 个节点上执行, i 和 j 满足 $0 \leq i \leq m-1, 0 \leq j \leq n-1$ 。

(2)目标函数设置。根据任务能耗模型,任务调度问题的目标函数为

$$\begin{aligned} \min(E) &= \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \alpha_i t_j + \sum_{i=0}^{n-1} \beta_i T_{\max} \\ \text{s. t. } &\begin{cases} \alpha_i = P_i - p_i \\ \beta_i = p_i \\ T_{\max} \geq T_i, i \in \{0, 1, \dots, n-1\} \end{cases} \end{aligned} \quad (11)$$

(3)新解产生规则。模拟退火算法的新解产生的规则会影响寻求最优值的效率。为减少运算,新解产生的规则应该尽可能简单。本文采用在邻域因子 N 内互换任务调度节点位置的方法产生新的调度序列。

(4)新解接受条件。新解接受准则是当新解的目标函数值小于当前解的目标函数值时,接受新解;否则以一定概率接受新解。新解接受条件为

$$(\Delta f < 0) \text{ 或 } (e^{-\Delta f/T} > \text{random}(0,1)) \quad (12)$$

(5)算法终止条件。当任务等待队列为空时,算法终止。

2.2 任务调度过程

根据调度问题的背景描述,通常情况下,任务个数 m 与节点个数 n 满足关系 $m \geq n$ 。因此,对于每一个空闲资源节点,调度过程需要经过若干轮才能完成 m 个任务的分配,任务调度过程如下。

步骤1 初始化资源列表 $R_0, R_1, \dots, R_{n-1}$ 、等待任务列表 $l_0, l_1, \dots, l_{m-1}$ 、模拟退火参数 U 和邻域因子 N 。

步骤2 根据 $(P_i - p_i) t_j$ 计算每个任务在当前节点 R_i 的能耗,取其中的最小值加入任务调度列表 S 。

步骤3 根据目标函数式(11)计算当前任务调度列表所有任务的总能耗。

步骤4 根据新解产生规则的设置,在邻域因子 N 内交换当前任务与其相邻任务在 S 中的位置,产生新的任务调度序列。

步骤5 根据目标函数式(11)计算新的任务调度列表所有任务的总能耗。

步骤6 计算步骤5和步骤3总能耗的差值,根据新解接受条件判断是否根据新的任务调度序列更新当前的任务调度列表。

步骤7 更新等待任务列表,并判断该列表中是否还有未调度的任务;如果所有任务调度完毕,任务调度过程结束;否则,根据上述步骤继续任务调度。

面向能耗优化的任务调度策略(simulated annealing based energy optimization oriented task scheduling strategy, SAEO)设计思想如下。SAEO策略首先初始化任务调度队列、临时任务调度队列以及退火控制参数 U 和解邻域空间大小参数 N 。其中, U 用于在新解产生时计算新解接受概率; N 用于控制新解产生的邻域空间大小。输出是任务调度列表 S 。初始化完毕后,将完成任务调度列表中任务的分配过程。当任务队列不为空的时候,计算每个任务在每个节点上运行的能耗,并通过列表存储计算结果。然后获取满足条件的调度任务,也就是将能耗最小的任务加入调度列表并计算当前调度列表中的任务总能耗。

接下来完成任务调度列表的模拟退火过程,首先交换任务队列两个元素的位置产生新序列;其次,计算新序列能耗及其与旧序列的能耗差;再次,根据初始参数配置的新解接受条件判断新序列在能耗优化的目标上是否比旧序列性能更好。如果性能更好,则根据旧序列更新老序列并退出当前搜索;否则继续搜索直至在当前任务邻域 N 内搜索完毕。接受新序列的条件由两部分组成,当新序列能耗值比旧序列能耗值小时,直接接受新序列;否则,以一定概率接受新序列。最后完成任务调度列表的更新,返回任务调度列表,完成任务调度过程。

2.3 算法复杂度分析

根据任务调度过程的描述可知,对于 m 个任务与 n 个节点的任务调度过程主要包括以下环节: n 个节点的扫描、 m 个任务的扫描和任务调度列表内 N 个任务的扫描。其中,邻域 N 内的任务扫描伴随着新任务调度序列的产生、能耗值比较及新序列接受条件的判断,这些步骤均在 $O(1)$ 时间内能够完成,因而整个算法的时间复杂度为 $O(mnN)$ 。当任务个数 m 和节点个数 n 较大时,算法效率可能极其

低下。因此,需要对算法进行一定的改进,降低算法的时间复杂度,提升其运行效率。

在任务的扫描环节,算法需要反复计算调度序列的能耗,存在着大量的重复运算。因此,在任务搜索过程中采用矩阵存储反复使用的中间数据,以空间换时间的方式改进算法的性能。矩阵采用二维线性表设计,提供 $O(1)$ 级别的数据查询服务。经过改进之后,算法在 $O(mn)$ 空间内存储中间数据,在 $O(mN)$ 时间内完成任务调度过程。

3 实验分析

为了验证能耗优化调度策略的有效性,本文采用浪潮 TS10000 高性能计算集群环境来搭建云渲染系统。渲染引擎使用 Blender 2.69,它是使用广泛的开源跨平台的三维绘图及渲染软件。测试数据选取电影“Big Buck Bunny”的部分镜头,该电影是 Blender 基金会第2部开放版权的动画电影。能耗测试仪器为 Watts'up,其额定电压为 250 V,额定电流为 15 A。

3.1 能耗性能分析

这里选取先进先出算法(first input first output, FIFO)和能耗感知的调度算法(energy aware MapReduce scheduling algorithm, EMRSA)^[11]进行能耗性能的对比分析。为了避免数据单位造成的混乱,采用 Max-Min 方法和 Z-score 方法对实验结果进行了归一化操作。本节实验及 SAEO 策略参数初始化如下:节点数为 40,作业数为 6,任务数为 100~1 000,新解搜索空间为 2,模拟退火因子为 10 000。

能耗性能的对比实验结果如图1所示。图中使用归一化方法对不同算法的能耗进行了处理,纵轴表示不同算法的能耗归一化值,值越大表示该算法的能耗越大,性能越低;横轴表示不同的任务规模。

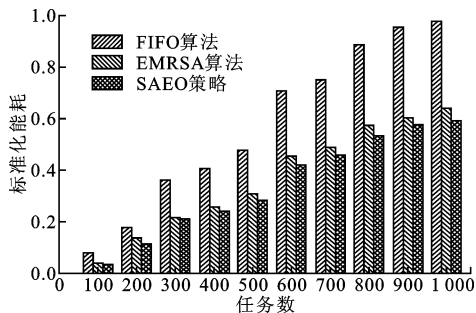


图1 多作业任务能耗测试

由图1可见,FIFO算法的性能表现最差,SAEO策略的性能表现最好。SAEO策略在多任务

条件下,性能比 FIFO 算法平均提升了 43.4%,比 EMRSA 算法平均提升了 6.7%。FIFO 算法的设计没有考虑任务能耗因素。EMRSA 算法使用了贪心思想,在满足用户时间要求的前提下最小化任务能耗,由于该算法仅着眼于当前作业的能耗最优,因而容易陷入局部最优化的困境。SAEO 策略使用了模拟退火思想,着眼于全局任务能耗最优,能够跳出局部最优的困境,针对多作业的调度在能耗性能上优于 EMRSA 算法。

3.2 任务规模扩展性分析

本节通过比较 EMRSA 算法和 SAEO 策略的相对性能提升,分析不同算法的适用场景。相对性能提升的计算如式(11)和式(12)所示。 E_{EMRSA} 和 E_{SAEO} 分别表示 EMRSA 算法和 SAEO 策略的能耗, O_{EMRSA} 和 O_{SAEO} 分别表示 EMRSA 算法和 SAEO 策略的运行时间。

$$E = \left| \frac{E_{EMRSA} - E_{SAEO}}{E_{EMRSA}} \right| \times 100\% \tag{13}$$

$$O = \left| \frac{O_{EMRSA} - O_{SAEO}}{O_{EMRSA}} \right| \times 100\% \tag{14}$$

由于 FIFO 算法没有考虑能耗因素,且该算法的能耗性能明显劣于其他两个算法,在此不与 FIFO 算法做性能的比较。SAEO 策略和 EMRSA 算法的相对性能提升比较如图 2 所示。

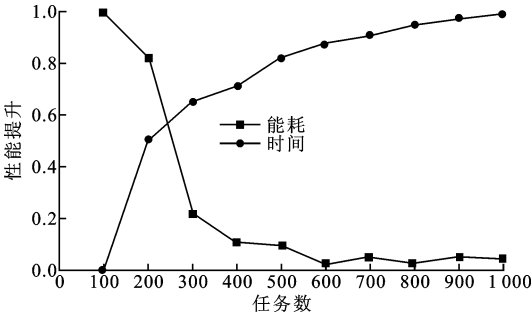


图 2 算法相对性能比较

由图 2 可见,SAEO 策略相对 EMRSA 算法的能耗改善比例伴随着任务规模的扩大快速降低,而执行时间增加比例伴随着任务规模扩大逐渐升高。其中,两个曲线在任务规模为 200 和 300 之间相交。因此,当任务规模小于 200 时,SAEO 策略在能耗方面的性能提升幅度大于其在时间性能方面的下降幅度,因而其算法性能整体上优于 EMRSA 算法;当任务规模大于 300 时,SAEO 策略在能耗方面的性能提升幅度小于其在时间性能方面的下降幅度,因而其算法性能整体上劣于 EMRSA 算法。上述对比分析结果表明,SAEO 策略在改善能耗性能方面

适用的最佳任务调度规模为 200~300。

3.3 集群规模扩展性分析

本节将分析不同算法在集群规模增加的情况下算法的扩展性问题。集群规模的扩大伴随着运行节点的个数增多,也意味着集群环境的复杂性增加。不同算法在进行任务调度时,其总能耗都随着集群规模的扩大不断增加。因此,在比较算法扩展性时,采用能耗效率作为衡量不同算法性能的指标

$$\eta = \left| \frac{E_w}{E_w + E_i} \right| \times 100\% \tag{15}$$

式中: E_w 表示集群运行任务的能耗; E_i 表示集群处于空闲状态的能耗。不同算法的能耗效率对比结果如图 3 所示。任务规模根据 3.2 节的结论选为 300,节点规模为 5~40。

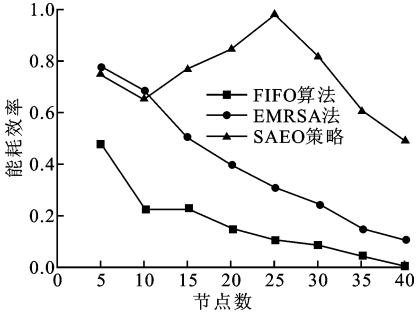


图 3 算法集群规模扩展性比较

由图 3 可见,随着节点规模的扩大,3 个算法的能耗效率总体上都呈下降趋势。因此,从能耗的角度,不能简单地通过扩大集群的规模提升其性能。另外,随着节点规模的增加,SAEO 算法下降的过程波动性比较大,这是由于 SAEO 算法采用模拟退火的思想搜索最优解的过程伴随着一定的概率。尽管 SAEO 算法的波动性比较大,算法稳定性相比 FIFO 和 EMRSA 算法不足,但是在能耗效率指标上,整体优于其他两种算法,这表明 SAEO 算法的集群规模扩展性更好。

4 结 论

云渲染在动漫影视制作领域应用越来越广泛,如何降低云渲染系统能耗受到越来越多的关注。本文针对云渲染系统的能耗优化问题,利用渲染应用帧与帧的并行性特征,综合考虑系统空闲节点和任务运行时能耗优化,建立渲染任务能耗模型,并基于该模型提出面向能耗优化的任务调度策略。该策略采用模拟退火算法的思想,通过局部能耗优化来实现全局能耗优化的目标。实验基于真实的多作业动画渲染案例,结果表明,本文提出的调度策略在能耗

优化方面比 FIFO 算法提升了 43.4%, 比 EMRSA 算法提升了 6.7%。通过分析, 与 EMRSA 算法的相对性能, 所提出的策略具有较好的任务规模扩展性, 同时在集群规模扩展性方面也优于其他两个算法。

参考文献:

- [1] BAHARON M R, SHI Q, LLEWELLYN-JONES D, et al. Secure rendering process in cloud computing [C] // Proceedings of the 2013 11th Annual International Conference on Privacy, Security and Trust. Piscataway, NJ, USA: IEEE, 2013: 82-87.
- [2] 罗亮, 吴文峻, 张飞. 面向云计算数据中心的能耗建模方法 [J]. 软件学报, 2014, 25(7): 1371-1387.
LUO Liang, WU Wenjun, ZHANG Fei. Energy modeling based on cloud data center [J]. Journal of Software, 2014, 25(7): 1371-1387.
- [3] LI Chao, ZHOU Ruijin, LI Tao. Enabling distributed generation powered sustainable high-performance data center [C] // Proceedings of the 19th IEEE International Symposium on High Performance Computer Architecture. Piscataway, NJ, USA: IEEE, 2013: 35-46.
- [4] LI Kebin. Energy efficient scheduling of parallel tasks on multiprocessor computers [J]. Journal of Supercomputing, 2012, 60(2): 223-247.
- [5] LEE Y C, ZOMAYA A Y. Energy efficient utilization of resources in cloud computing system [J]. Journal of Supercomputing, 2012, 60(2): 268-280.
- [6] ZONG Ziliang, MANZANARES A, RUAN Xiaojun, et al. EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters [J]. IEEE Transactions on Computers, 2011, 60(3): 360-374.
- [7] MAHESHWARI N, NANSURI R, VARMA V. Dynamic energy efficient data placement and cluster reconfiguration algorithm for MapReduce framework [J]. Future Generation Computer System, 2012, 28(1): 119-127.
- [8] PIETRI I, SAKELLARIOU R. Energy-aware workflow scheduling using frequency scaling [C] // Proceedings of the 2014 43rd International Conference on Parallel Processing Workshops. Piscataway, NJ, USA: IEEE, 2014: 104-113.
- [9] 朱晓敏, 贺川, 王建江, 等. 异构计算系统中弹性节能调度策略研究 [J]. 计算机学报, 2012, 35(6): 1313-1326.
ZHU Xiaomin, HE Chuan, WANG Jianjiang, et al. An elastic energy-aware scheduling strategy for hetero-

geneous computing systems [J]. Chinese Journal of Computers, 2012, 35(6): 1313-1326.

- [10] 谭一鸣, 曾国荪, 王伟. 随机任务在云计算平台中能耗的优化管理方法 [J]. 软件学报, 2012, 23(2): 266-278.
TAN Yiming, ZENG Guosun, WANG Wei. Policy of energy optimal management for cloud computing platform with stochastic tasks [J]. Journal of Software, 2012, 23(2): 266-278.
- [11] MASHAYEKHY L, NEJAD M M, GROSU D, et al. Energy-aware scheduling of MapReduce jobs [C] // Proceedings of the IEEE International Congress on Big Data. Piscataway, NJ, USA: IEEE, 2014: 32-39.
- [12] GUZEK M, PECERO J E, DORRONSORO B, et al. Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems [J]. Applied Soft Computing, 2014, 24: 432-446.
- [13] SOUSA T, MORAIS H, VALE Z, et al. Intelligent energy resource management considering vehicle-to-grid: a simulated annealing approach [J]. IEEE Transactions on Smart Grid, 2012, 3(1): 535-542.

[本刊相关文献链接]

- 周墨颂, 朱正东, 董小社, 等. 采用资源划分的云环境下 Hadoop 资源许可调度方法. 2015, 49(8): 69-74. [doi: 10.7652/xjtuxb201508012]
- 樊源泉, 伍卫国, 许云龙, 等. MapReduce 环境中的性能特征能耗估计方法. 2015, 49(2): 14-19. [doi: 10.7652/xjtuxb201502003]
- 马莉, 唐善成, 王静, 等. 云计算环境下的动态反馈作业调度算法. 2014, 48(7): 77-82. [doi: 10.7652/xjtuxb201407014]
- 庄威, 桂小林, 林建材, 等. 云环境下基于多属性层次分析的虚拟机部署与调度策略. 2013, 47(2): 28-32. [doi: 10.7652/xjtuxb201302005]
- 李健, 黄庆佳, 刘一阳, 等. 云计算环境下的大规模图状数据处理任务调度算法. 2012, 46(12): 116-122. [doi: 10.7652/xjtuxb201212020]
- 曹仰杰, 钱德沛, 伍卫国, 等. 一种面向多处理器系统的在线低功耗调度算法. 2010, 44(8): 15-19. [doi: 10.7652/xjtuxb201008004]
- 康雨, 闫相国, 郑崇勋, 等. 任意可分负载的多轮调度算法. 2009, 43(8): 125-129. [doi: 10.7652/xjtuxb200908025]
- 王毅, 张德运, 梁涛涛. 无线传感器网络分区能耗均衡的非均匀分簇算法. 2008, 42(4): 389-394. [doi: 10.7652/xjtuxb200804002]

(编辑 武红江)