doi:10.11959/j.issn.1000-436x.2015017

面向数据密集型工作流的能耗感知调度策略

肖鹏[1]，胡志刚[2]，屈喜龙[1]
（1. 湖南工程学院 计算机与通信系，湖南 湘潭 411104；2. 中南大学 软件学院，湖南 长沙 410083）

摘 要：随着数据中心规模的扩大，高能耗问题已经成为高性能计算领域的一个重要问题。针对数据密集型工作流的高能耗问题，提出通过引入“虚拟数据访问节点”的方法来量化评估工作流任务的数据访问能耗开销，并在此基础上设计了一种“最小能耗路径”的启发式策略。在经典的 HEFT 算法和 CPOP 算法基础上，通过引入该启发式策略设计并实现了 2 种具有能耗感知能力的调度算法（HEFT-MECP 和 CPOP-MECP）。实验结果显示，基于最小能耗路径的启发式调度算法能有效降低数据访问操作的能耗开销，在面对大型的数据密集工作流任务时，该启发式调度策略体现了较好的适应性。

关键词：工作流；能耗；启发式策略；云计算
中图分类号：TP393 **文献标识码**：A

Energy-aware scheduling policy for data-intensive workflow

XIAO Peng[1], HU Zhi-gang[2], QU Xi-long[1]
(1. Department of Computer and Communication, Hunan Institute of Engineering, Xiangtan 411104, China;
2. School of Software, Central South University, Changsha 410083, China)

Abstract: With the increasing scale of data centers, high energy consumption has become a critical issue in high-performance computing area. To address the issue of energy consumption optimization for data-intensive workflow applications, a set of virtual data-accessing nodes are introduced into the original workflow for quantitatively evaluating the data-accessing energy consumption, by which a novel heuristic policy called minimal energy consumption path is designed. Based on the proposed heuristic policy, two energy-aware scheduling algorithms are implemented, which are deprived from the classical HEFT and CPOP scheduling algorithms. Extensive experiments are conducted to investigate the performance of the proposed algorithms, and the results show that they can significantly reduce the data-accessing energy consumption. Also, the proposed algorithms show better adaptive when the system is in presence of large-scale workflows.

Key words: workflow; energy consumption; heuristic policy; cloud computing

1 引言

早期的能耗优化技术研究大多集中在嵌入式系统领域，目标一般是延长电池使用时间[1]。随着高性能计算平台在商业领域的广泛，数据中心的能耗开销成为系统运营的主要成本之一，面向数据中心的能耗优化问题因此成为当前亟待解决的课题[2,3]。

在高性能计算领域，能耗优化技术一般是在“能耗/性能”两者之间进行权衡，最典型的策略就是关闭处于空闲状态的设备。京都大学高性能计算中心在 2004 年~2007 年的统计报告显示[4]，仅仅采用不定期地关闭空闲节点的策略就实现了 39%的能耗节约。这种“关闭/启动”节能方式的主要缺点是：

收稿日期：2013-08-23；**修回日期**：2013-12-01
基金项目：国家自然科学基金资助项目(61402163, 61272148)；湖南省教育厅科学研究基金资助项目(13B015)；湖南省科技计划项目基金资助项目(2012GK3075)；湖南省自然科学基金资助项目(13JJ9022)
Foundation Items: The National Natural Science Foundation of China (61402163, 61272148); The Scientific Research Fund of Hunan Provincial Education Department (13B015)；Provincial Science & Technology Plan Project of Hunan (2012GK3075); Hunan Provincial Natural Science Foundation of China (13JJ9022)

设备从关闭状态到启动状态需要一定的时间，从而导致系统总体性能损失过大[2,5]。近期，各种物理设备已经开始支持不同能耗的多工作模式，包括处理器的DVFS (dynamic voltage and frequency scaling)技术[6,7]，磁盘的转速可调技术[8]等。这些技术的主要特点是设备可以在不同能耗模式之间动态切换，从而降低关闭设备所造成的性能开销，以此实现“能耗/性能”的平衡。

近期，研究者开始对系统能耗的分布和影响因素进行了大量的研究，相关研究结果显示：1) 任务行为特性（数据密集型/计算密集型）对系统能耗分布具有决定性的影响[8~10]；2) 任务结构特性（并行特征/串行特征）对能耗开销具有重要影响[11,12]。这些研究结论指出，从“任务结构和行为特性”的角度来研究能耗优化技术是实现细粒度能耗优化的重要方法之一。在分布式计算领域，数据密集型工作流是科学研究和工程计算中最为典型的一种任务类型，其数据密集的特征决定了数据中心的能耗分布偏向磁盘能耗[8,13]，其结构特征对数据传输延迟的影响又进一步影响了处理器和存储磁盘的能效[14]。因此，本文主要研究数据密集型工作流的能耗模型和调度模型，通过引入启发式策略来设计具有能耗感知能力的调度算法。

2 相关研究

针对I/O操作密集型任务的能耗问题，早期的研究者首先从分布式存储系统的体系结构方面进行了若干探索。例如，Khargharia等人[15]设计了一个4层能耗管理模型，其底层采用“实时监控—反馈”机制来调整物理资源的工作模式，中间层则采用约束优化技术来匹配能耗与任务QoS需求之间的关系，最上层则为能耗优化策略提供相应的决策参数。这些存储体系结构从宏观层面上为能耗优化提供了基础，但面对系统负载动态变化的情况，单靠改进存储体系结构是无法有效实现“自适应”和“细粒度”的能耗优化与控制。

为此，研究者开始针对I/O操作的模式特征进行了大量研究，并提出了若干面向能耗优化的分布式存储技术。例如，Garg等人[8]就提出了一种基于马尔科夫模型的I/O访问模式预测技术，通过分析磁盘在不同工作状态之间的转换迁移日志来分析未来磁盘的可能工作状态，从而预先提出相应的能耗优化方案；针对数据访问操作的局部性特征，王桂彬等人[16]分析了OpenMP并行程序中循环结构的能耗下界，并提出一种基于整数规划技术的能耗最优策略。以上基于I/O模式的能耗优化技术一般归类为“静态能耗优化”策略。

考虑到分布式系统的动态性和异构性，“动态能耗优化”技术逐渐成为近期的研究热点。例如，Niyato等人[17]提出了一种面向节能的批任务调度算法，该算法首先利用马尔科夫模型来描述分布式资源池的动态能耗过程，然后通过状态迁移矩阵将批任务调度问题归结为约束规划问题。但该算法并未将任务执行性能作为约束条件，因此无法确定调度方案所导致的性能损失。Rizvandi等人[7]将传统的Max-Min算法与DVFS技术相结合，提出一种针对DAG任务的MMF-DVFS调度算法，该算法与Max-Min算法的唯一区别在于其启发函数为DAG任务的总能耗而不是调度长度。Lee等人[18]提出了2种能耗感知的DAG启发式调度算法（ECS和ECS+idle），其核心思想是利用DVFS技术对处理器工作电压进行迭代调整，并保证每次调整后的任务执行长度保持不变，2种算法的主要区别在于ECS只考虑处理器在活动状态下的能耗，而ECS+idle则将处理器空闲状态下的能耗计入总能耗。

本文所提出的调度策略属于“动态能耗优化技术”。从研究对象而言，文献[8]和文献[18]与本文研究最为接近，但这些研究的基本技术路线都是在传统DAG调度算法中加入DVFS技术，通过“频率缩放”技术来降低处理器的能耗开销，其目标是降低计算密集型任务的能耗或提高其能效。而本文所提调度策略主要针对数据密集型工作流，考虑到这类工作流的能耗分布偏向磁盘能耗，本文主要通过分析数据访问操作时的能耗开销及其影响因素来设计启发式调度策略。

3 问题描述与定义

数据密集型工作流中子任务之间的数据传输量往往很大，其传输过程需要借助数据中心的存储节点，而不能简单地描述为计算节点之间的直接数据传输，其执行过程如图1所示。为方便下文描述，此处首先给出若干定义及相关说明。

定义1 设目标系统包括N个计算节点和M个数据存储节点，分别表示为集合$\{C_1,C_2,\cdots,C_N\}$和$\{S_1,S_2,\cdots,S_M\}$。

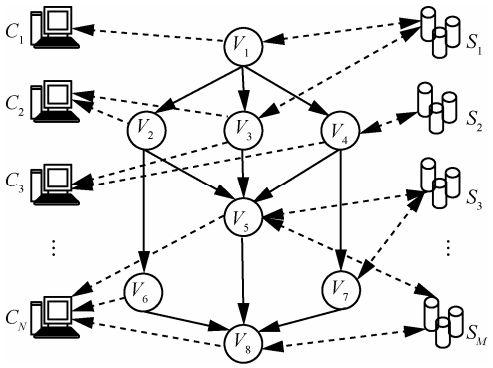


图 1　数据密集型工作流的执行过程

定义 2　工作流任务表示为有向无环图 $G=<V, W>$，其中 $V=\{V_1,V_2,\cdots,V_n\}$为工作流的子任务集合，每个子任务 V_i由三元组$<I_i, D_i^{\text{in}}, D_i^{\text{out}}>$表示，$I_i$为 V_i的计算工作量（以指令数来衡量），D_i^{in}和 D_i^{out}分别为 V_i的输入/输出数据；$W=\{W_{i,j}|<V_i, V_j>\in V\times V\}$为工作流的有向边集合，其数值代表的 V_i和 V_j之间的数据传输开销。

定义 3　工作流 DAG 的初始任务节点和终止任务节点分别记为 V_{init}和 V_{exit}；任意子任务 V_i的前驱任务节点集合记为 $Prev(V_i)$，后继任务节点集合记为 $Succ(V_i)$。

定义 4　计算节点到存储节点之间通信带宽用 $N\times M$ 型矩阵 $\boldsymbol{B}$ 表示，其中的矩阵元素 $B_{i,j}(1\leqslant i\leqslant N, 1\leqslant j\leqslant M)$表示 C_i到 S_j之间的通信带宽。

定义 5　任务调度方案为 $M:V\times C\times S\rightarrow\{0,1\}$，表示子任务集合到计算节点和存储节点集合的一个有效映射，其矩阵元素 $M_{i,i',i''}$表示子任务 V_i 调度到计算节点 $C_{i'}$上执行，其输出数据存储到节点 $S_{i''}$上。

定义 6　给定调度方案 M，工作流任务顺利执行完毕的总能耗 $E(G, M)=\sum E_i(M_{i,i',i''})$，其中，$E_i(M_{i,i',i''})$为执行子任务 V_i所需的能耗。

从图 1 所示的工作流执行过程可以看出，执行 V_i所需的能耗 $E_i(M_{i,i',i''})$包括计算任务能耗 $E^C{}_i(M_{i,i',i''})$和数据存储能耗 $E^S{}_i(M_{i,i',i''})$ 2 个部分。综合以上关于数据密集型工作流的定义和能耗计算模型可知，获得能耗最优化调度方案的问题可以描述为以下最优规划问题

$$\begin{cases}\min\ E(G,M)=\sum_{i=1}^{n}\left(E_i^C(M_{i,i',i'})+E_i^S(M_{i,i',i''})\right)\\ \text{满足 } M_{i,i',i''}\in V\times C\times S\rightarrow\{0,1\}\end{cases}\quad(1)$$

由于 M 的解空间尺寸为 2^{n+N+M}，以上最优规划问题显然属于 NP-Complete。为此，将首先分析数据密集型工作流的能耗模型，并提出一种具有能耗感知能力的启发式策略用于调度算法的设计和实现。

4　能耗感知调度模型与算法设计

4.1　工作流任务的能耗模型

依据 CMOS 电路的功耗特性，计算节点的功耗为工作频率 f 的函数。

$$P^c=\beta f^{\alpha}\quad(2)$$

其中，β 和 α 都是与处理器工艺相关的常数，α 一般取值区间为[2,3]。因此，在给定调度方案 $M_{i,i',i''}$下，计算任务能耗 $E_i^C(M_{i,i',i''})$的计算公式为

$$E_i^C(M_{i,i',i''})=P_{i'}^c\frac{I_i}{f_{i'}}=\beta_{i'}f_{i'}^{(\alpha-1)}I_i\quad(3)$$

数据存储能耗包括输入/输出数据的能耗。由于网络通信的延迟要远远大于数据节点的网络端口到其磁盘之间延迟，因此，$E^S{}_i(M_{i,i',i''})$的计算公式可以表示为

$$E_i^S(M_{i,i',i''})=\sum_{j\in Prev(v_i)}\left(P_{j''}^s\frac{W_{j,i}}{B_{i',j''}}\right)+\sum_{k\in Succ(v_i)}\left(P_{i''}^s\frac{W_{i,k}}{B_{i',i''}}\right)\quad(4)$$

其中，前半部分为 V_i从其前驱任务中（即数据节点 $S_{j''}$中）获得输入数据所需的能耗，后半部分为 V_i输出数据到其存储节点 $S_{i''}$而所需的能耗，P_i^s为存储节点 S_i在全速状态下的能耗函数，一般与其最高磁盘转速成正比，本文视为与设备工艺相关的常数。

图 1 所示的数据密集型工作流的一个显著特征是 DAG 中边的权值（即通信时间开销）是由前驱节点的输出开销和后继节点的输入开销 2 部分组成，数据节点主要充当存储访问中介的作用。为清晰描述调度模型，本文为在原 DAG 的基础上引入一组“虚拟数据访问节点”，表示每个子任务节点(V_{exit} 节点除外)的数据输出操作，记为$\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$。图 2 所示为引入“虚拟数据访问节点”后 DAG 结构转变。

由于所有“虚拟数据访问节点”的计算工作量为 0，且其最终都被映射到数据存储节点集合$\{S_1,S_2,\cdots,S_M\}$，因此转换后的 DAG 与原工作流的逻辑结构等价。在引入“虚拟数据访问节点”后，数据访问的能耗开销完全可以由 V_i^*节点以及其相关联的有向边来表示，结合式(4)可知，V_i^*的数据存储能耗 $E^*{}_i(M_{i,i',i''})$为

$$E_i^*(M_{i,i',i''}) = P_{i''}^s\left(\frac{D_i^{\text{out}}}{B_{i',i''}} + \sum_{V_j \in Succ(V_i)} \frac{W_{i,j}}{B_{j',i''}}\right) \quad (5)$$

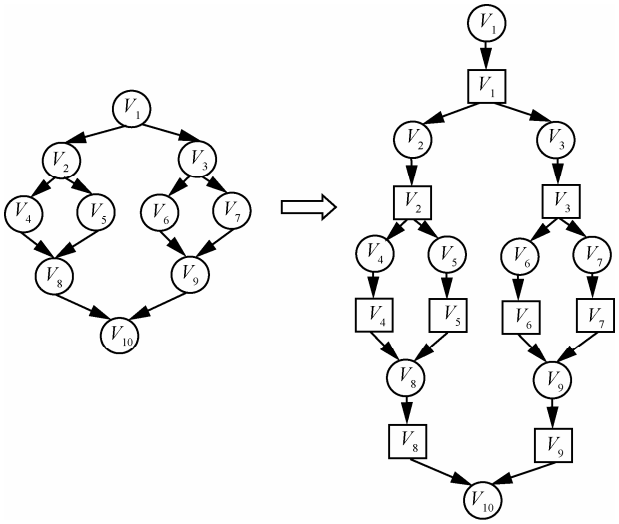


图 2 引入“虚拟数据访问节点”的 DAG 结构

与式(4)相比，式(5)描述的数据存储能耗与计算任务节点完全隔离，更清晰地反映了数据密集型工作流的执行过程。因此，本文将用式(5)来描述工作流的数据存储能耗，即工作流的整体能耗式为

$$E(G,M) = \sum_{i=1}^{n} E_i^C(M_{i,i',i''}) + \sum_{i=1}^{n-1} E_i^*(M_{i,i',i''}) \quad (6)$$

由图 2 所示的工作流结构可知，若不对计算任务节点的能耗进行优化，而只考虑数据存储能耗优化问题，则只需简单地将{ $V_1^*,V_2^*,\cdots,V_{n-1}^*$}中的元素一一映射到{$S_1,S_2,\cdots,S_M$}，并确保 $\forall V_i^*$均满足 $\min\{E_i^*(M_{i,i',i''})\}$即可，求解该问题的时间复杂度为 $O((n-1)M)$。若需要同时考虑计算能耗、数据存储能耗和调度性能，则问题的难点在于给定调度方案，计算总能耗$\sum E_i^C(M_{i,i',i''})$、数据存储总能耗$\sum E_i^*(M_{i,i',i''})$以及调度性能（如 Makespan 指标）无法同时达到最优。下文的核心内容就是给出一种启发式调度算法，在保证调度性能的前提下，同时优化工作流任务的执行能耗。

4.2 数据密集型工作流的能耗感知调度算法

设 $M_{i,i',i''}$为 V_i的一个调度方案，表示 V_i被调度到计算节点 $C_{i'}$上执行，且其后继的虚拟数据访问节点 V_i^*被映射到数据节点 $S_{i''}$上。在引入“虚拟数据存储节点后”，有向边的权值（数据通信时间）按式(7)计算。

$$\omega_{i',i''} = \begin{cases} \dfrac{D_i^{\text{out}}}{B_{i',i''}}, \text{若} W_{i,j} \text{为 } V_i^* \text{的入度边} \\ \dfrac{W_{i,j}}{B_{j',i''}}, \text{其他} \end{cases} \quad (7)$$

因此，子任务 V_i的最早启动时间 $EST(M_{i,i',i''})$和最早完成时间 $EFT(M_{i,i',i''})$的计算公式如下

$$EST(M_{i,i',i''}) = \begin{cases} 0, \ V_i = V_{init} \\ \max\limits_{V_j \in Prev(V_i)} \left\{ EFT(M_{j,j'',j''}) + \dfrac{D_j^{out}}{B_{j',j''}} + \dfrac{W_{j,i}}{B_{i',j''}} \right\}, \text{其他} \end{cases} \quad (8)$$

$$EFT(M_{i,i',i''}) = EST(M_{i,i',i''}) + \frac{I_i}{f_{i'}} \quad (9)$$

在传统的面向“Makespan 最小化”的启发式算法中，*EST* 和 *EFT* 指标被直接用于计算各个任务节点的优先级（b-level 或 t-level）。本文主要关注数据密集型工作流的能耗优化调度，因此任务的主要能耗开销将集中在数据存储操作上。为此，本文引入了一个“最小能耗路径”（MECP, minimal energy consumption path）的启发式指标，用于分析“虚拟数据访问节点”在工作流执行过程中的能耗情况，其定义如下

$$MECP(V_i^*) = \begin{cases} E_i^*(M_{i,i',i''}), \ V_i^* = V_{\text{init}} \\ E_i^*(M_{i,i',i''}) + \min\limits_{V_j^* \in Prev(V_i^*)} \left\{ MECP(V_i^*) \right\} \end{cases} \quad (10)$$

MECP 指标所表示的是当前“虚拟数据访问节点”到起始节点之间总能耗最小的路径。由于 *MECP* 指标只针对“虚拟数据访问节点”的能耗，因此原 DAG 的子任务{$V_1,V_2,\cdots,V_n$}的优先级可直接采用传统启发式调度策略的计算方法，而{$V_1^*,V_2^*,\cdots,V_{n-1}^*$}的优先级需做如下修正。

$$rank_t(V_i^*,S_k) = \frac{\max\limits_{V_j \in Prev(V_i^*)} \left\{ rank_t(V_i^*,S_j) + \dfrac{D_j^{out}}{B_{j,k}} + \dfrac{W_{j,i}}{B_{j,k}} \right\}}{MECP(V_i^*)} \quad (11)$$

$$rank_b(V_i^*,S_k) = \frac{\max\limits_{V_j \in Succ(V_i^*)} \left\{ rank_b(V_i^*,S_j) + \dfrac{D_j^{out}}{B_{j,k}} + \dfrac{W_{j,i}}{B_{j,k}} \right\}}{MECP(V_i^*)} \quad (12)$$

其中，式(11)用于计算非起始节点的 t-level 与 *MECP* 指标的比值，即能耗越大则 V_i^*的优先级越低；式(12)

用于计算非终止节点的 b-level 与 *MECP* 指标的比值，即能耗越大则 V_i^*的优先级越低。

基于以上“虚拟数据访问节点”的优先级定义，本文提出 2 种具有能耗感知能力的 DAG 调度算法：HEFT-MECP 和 CPOP-MECP。2 种算法分别基于经典的 HEFT 算法和 CPOP 算法[19]，关键的区别在于：对 DAG 中的“虚拟数据访问节点”采用式(11)和式(12)进行优先级计算，目标在于最小化数据访问过程中的能耗开销。具体的算法实现如下所示。

HEFT-MECP 算法

输入：

1) 任务 DAG : $G = <V, W>$

2) 节点之间的带宽矩阵: $\boldsymbol{B}$

输出：

1) 调度方案: $\boldsymbol{M}$

Begin

1) 通过插入虚拟节点$\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$，将$G=<V, W>$转化为 $G^* =<<V,V^*>, W>$;

2) 从终止节点开始反向计算各个任务节点的 $rank_b$ 值;

3) 按 $rank_b$ 非递增顺序排序调度队列中的各个任务节点;

4) **while** 调度队列不为空 **do**

5) 选择队首节点 V_i作为调度任务；

6) 若 $V_i\in\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$, 则依据式(11)计算所有 $S_j \in \{S_1,S_2,\cdots,S_M\}$的 $rank_t(V_i, S_j)$值，并将 V_i调度调满足 $\max\{rank_b+ rank_t\}$条件的节点；

7) 否则，依据式(9)计算所有 $C_j\in \{C_1,C_2,\cdots, C_N\}$ 的 $EFT(M_{i,k,j})$, 并将 V_i 调度调满足 $\max\{EFT(M_{i,k,j})\}$条件的节点上;

8) **end while**

End

CPOP-MECP 算法

输入：

1) 任务 DAG : $G = <V, W>$

2) 节点之间的带宽矩阵: $\boldsymbol{B}$

输出：

1) 调度方案: $\boldsymbol{M}$

Begin

1) 通过插入虚拟节点$\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$，将$G=<V, W>$转化为 $G^* =<<V,V^*>, W>$;

2) 从终止节点开始反向计算各个任务节点的 $rank_b$ 值;

3) 从初始节点开始计算各个任务节点的 $rank_t$ 值;

4) 将的 G^*关键路径存入 L_{cp};

5) 找到满足条件 $\min\{\sum_{V_i\in L_{cp}} E(M_{i,k,j})\}$的 C_k 和 S_j;

6) 将所有关键路径上的计算任务节点 V_i 调度到 C_k;

7) 将所有关键路径上的数据访问节点 V_i^*调度到 C_k;

8) **while** 调度队列不为空 **do**

9) 选择满足条件$\max\{rank_b\}$的节点 V_i作为调度任务；

10) 若 $V_i\in\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$, 则将 V_i调度到满足条件 $\max\{ rank_b+ rank_t \}$的节点 S_k;

11) 否则, 以插入调度法将 V_i 调度到满足条件$\min\{EFT\}$的节点上;

12) 更新调度队列中的剩余任务的 $rank_b$ 值;

13) **end while**

End

4.3 算法分析与讨论

由 4.1 节的描述可知，“虚拟数据访问节点”总是被映射到存储节点集合$\{S_1,S_2,\cdots,S_M\}$，而该集合与计算节点集合$\{C_1,C_2,\cdots,C_N\}$不存在交集。因此，在 HEFT-MECP 算法和 CPOP-MECP 算法中，原 DAG 中的计算任务节点$\{V_1,V_2,\cdots,V_n\}$仍依据传统的 HEFT 算法和 CPOP 算法进行调度。

在 HEFT-MECP 算法中，“虚拟数据访问节点”$\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$采用逐一映射的方式进行调度，其启发函数采用的是修改后的 b-level 和 t-level。由 MECP 指标的定义可知，算法的每轮调度都选择数据存储能耗最小的数据存储节点来映射 V_i^*。结合式(5)可以看出，除了存储节点自身的功耗特性外，影响数据存储能耗的关键因素是数据存储量与对应的网络带宽之间的比值。因此，那些与其他计算节点之间具有较大带宽连接的存储节点将被优先考虑。

CPOP-MECP 算法首先将关键路径上的所有任务调度到单一计算节点和存储节点上。在经典的 CPOP 算法中，关键路径上任务之间的通信开销是被忽略的。由于本文考虑的 DAG 都需要借助独立的存储节点进行数据中转，因此采用 CPOP-MECP 算法时，关键任务之间通信开销主要由2个被优先选择出来的计算节点和存储节点之间带宽决定（如 CPOP-MECP 算法中的步骤 4）~步骤 7）所示）。由于本文主要关注数据存储过程中的能耗优化，因此在映射关键路径

时，直接采用了式(5)所示的能耗指标作为启发函数，这与传统的 CPOP 算法存在不同。

需要指出的是，HEFT-MECP 和 CPOP-MECP 算法在调度$\{V_1^*,V_2^*,\cdots,V_{n-1}^*\}$时，并未采用“插入调度策略”，原因在于相对计算节点而言，数据存储节点具有较高的并发处理能力，即当不同的“虚拟数据访问节点”调度到同一个 S_i 上时，算法无需保证这些任务节点的先后关系。最后，由于插入了 n–1 个“虚拟数据访问节点”，原 DAG 中将增加 n–1 条边，因此 2 个算法的时间复杂度均为 $O((e+n-1)\cdot(N+M))$，其中，e 和 n 分别为原 DAG 中的边和子任务数目，N 和 M 为计算资源节点和数据存储节点数。

5 实验与性能分析

为分析和评估 HEFT-MECP 算法和 CPOP-MECP 算法的性能表现，组织了 2 组实验：第一组实验用随机生成的 DAG 任务在 CloudSim[20]平台上进行测试和分析；第二组实验用真实工作流在实际系统上进行测试和分析。为了综合评估算法的能耗优化效果，实验结果除了与传统的 HEFT 和 CPOP 2 种不支持能耗优化的算法进行比较外，还选择了 MMF-DVFS[8]这种具有“能耗感知”能力的算法用于实验对比分析。

5.1 随机 DAG 的实验分析

实验首先采用随机生成的DAG作为任务负载，在随机生成 DAG 时，其中 DAG_{size} 表示任务节点数，DAG_{CCR} 表示工作流的通信计算比（CCR, communication computation ratio）。实验平台中包含 10 个高性能计算集群和 7 个数据存储中心。本次实验主要考察的性能指标包括调度长度（makespan）和总能耗及其分布（energy distribution）。

图 3 所示结果显示，不具备能耗感知能力的调度算法（HEFT 和 CPOP）能获得较好的 makespan 指标，而 MMF-DVFS 算法在该指标方面的表现最差，尤其是当 *CCR* 较小时其调度长度超过 HEFT 约 43%，而 HEFT-MECP 的调度长度与 HEFT 的差别则明显较小。2 个较为明显的实验结果是：1) 调度长度随任务规模的增加而增加，且存在一个明显的拐点，即当任务规模超过 150 后，所有调度算法对应的 makespan 都呈加速增加的趋势；2) *CCR* 参数与 makespan 呈正相关性，即随着任务的数据通

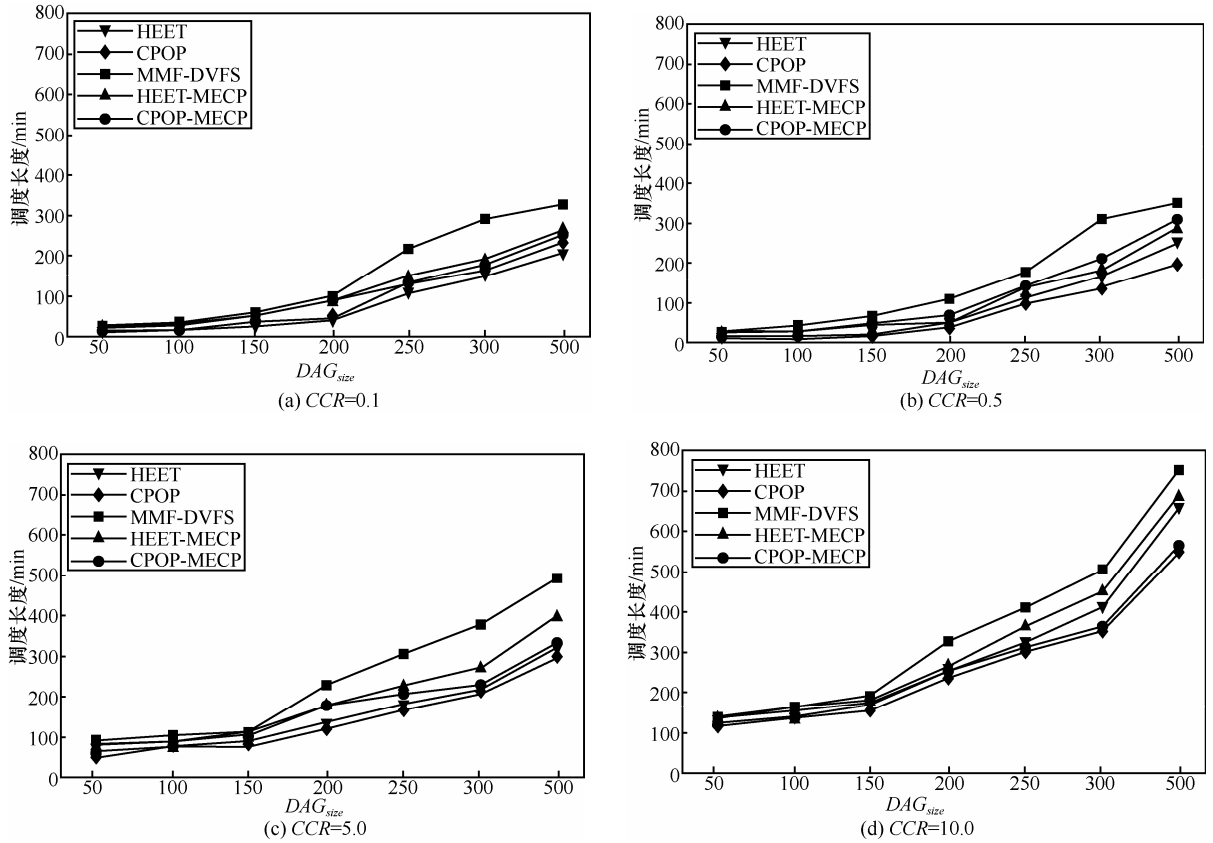


图 3 不同 *CCR* 下的任务调度长度

信开销逐步增加，其对应的调度长度也显著增加。

第一个实验结果与实验的平台配置相关，当任务规模较小时，由于系统可用资源相对丰富，各种调度算法都能依据简单的启发式技术来执行调度决策。此时，影响 makespan 的主要因素不是算法本身，而是可用资源总量。因此，当任务规模较小时，各种调度算法的 makespan 指标的差异很小；当任务规模逐步增加时，不同算法的性能差异开始增大。第二个实验结果集中体现了任务自身特征对调度效率的影响。在本实验环境下，发现采用相同的调度算法对相同规模的 DAG 进行调度时，*CCR* 参数成为影响调度长度的主要因素。

为了准确分析调度算法对能耗的影响，在实验中引入了 2 个统计指标：无效能耗（IEC, ineffective energy consumption）、计算能耗（CEC, computation energy consumption）和数据访问能耗（DAEC, data access energy consumption）。在不影响实验分析的前提下，实验做了如下设定：除 MMF-DVFS 算法外，在执行其余 4 种调度算法时，处理器在空闲状态下的能耗为工作状态下的 40%；存储设备在空闲状态下的能耗为工作状态下的 60%。能耗统计以调度算法开始为起点，以任务执行完毕为终点。实验统计了上文所有实验过程中的能耗情况，由于篇幅所限，只列出了 DAG_{size}=500 时的实验数据，如图 4 所示。

实验结果显示，随着 *CCR* 参数值的增加，各种调度算法的总能耗都呈增加趋势。导致这种现象的原因是 *CCR* 的增加会导致调度长度显著增加，由此增加了实验所用的时间，从而导致总能耗的增加。当 *CCR* 较小时，DAEC 在总能耗中的比重很小，而 CEC 所占的比重则相对大很多。因此，具有处理器能耗优化能力的 MMF-DVFS 的性能表现最

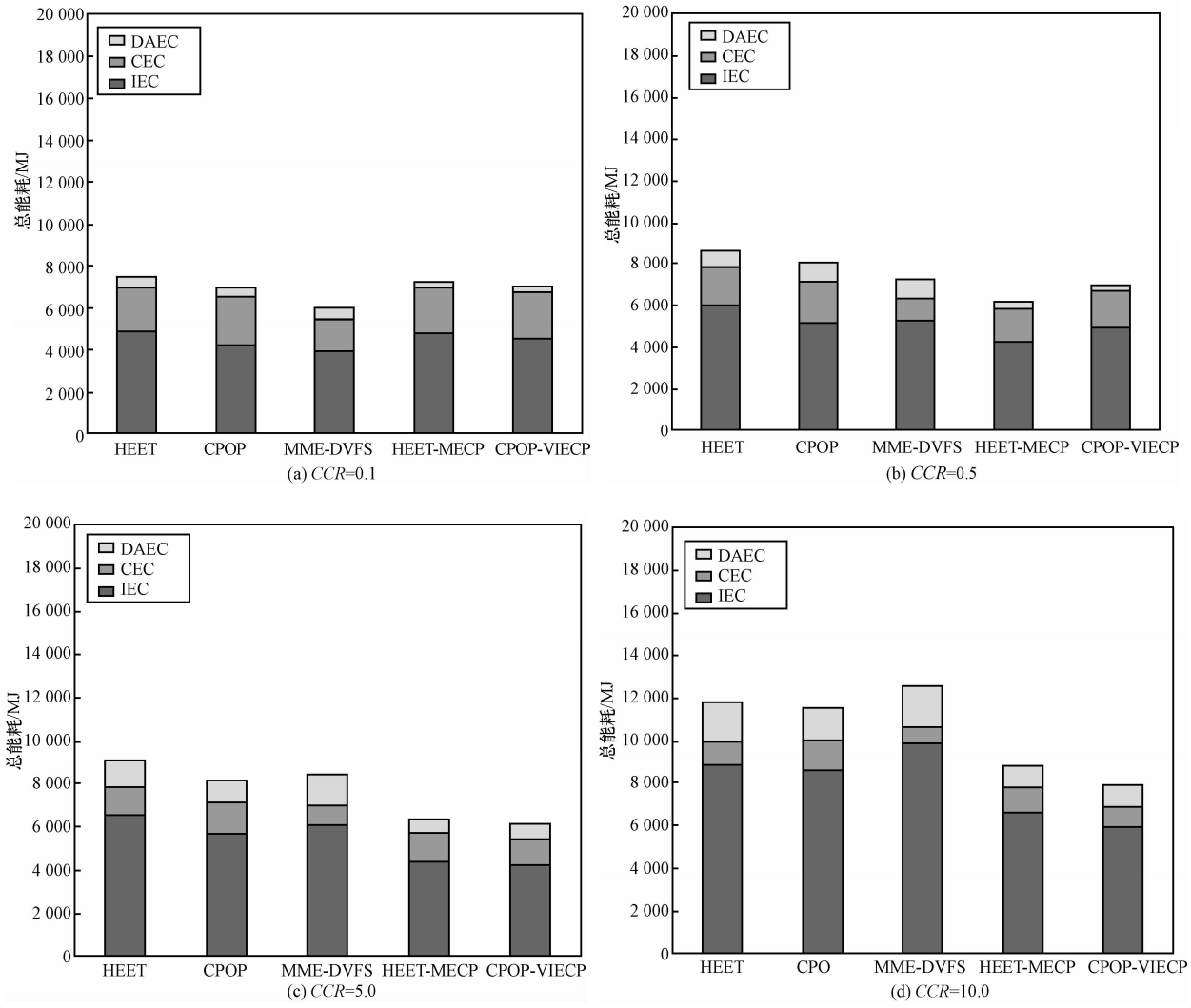


图 4　不同 *CCR* 下的总能耗及其分布(DAG_{size}=500)

优。随着 *CCR* 的增加，DAEC 的比重逐渐增大，而 CEC 的比重则逐步减小。在采用 HEFT 和 CPOP 算法时，DAEC 指标的增加更为明显。因此，它们对数据密集型 DAG 的数据访问能耗具有较好的优化作用。从总能耗而言，当 *CCR* 参数超过 5.0 后，HEFT-MECP 和 CPOP-MECP 的节能表现要明显优于 HEFT 和 CPOP 算法。这种总能耗的优化效果一部分来自于 DAEC 指标的降低，另一部分则来自 IEC 指标的降低。

5.2 实际工作流的实验分析

本次实验以 WIEN2K[21]工作流为测试对象，其问题规模由 2 个并行分支的大小 n 决定，本实验设定问题规模从 100 逐步增加到 400。实验平台包括 3 个同构计算集群和 9 个数据存储节点。其中计算集群的处理器类型分别为 Pentium M、AMD Crusoe TM-5800 和 AMD Turion MT-340；数据存储节点包括 2 个 IBM Ultrastar RAID 和 7 个 WDC iSCSI Sever。

实验结果如图 5 所示，本实验与模拟实验的一个显著差异在于系统无效能耗大幅度降低，而数据访问能耗在总能耗中的比重明显增加。其原因在于：模拟实验中的目标系统规模很大，因此很多资源在模拟过程中均处于非饱和状态，从而导致 IEC 指标很大；在本次实验中，目标系统中很多资源在实验过程中均处于过饱和状态，因此降低了系统无效能耗开销。

在实验过程中，计算能耗的开销大约占总能耗的 12%~30%，绝大部分的能耗集中在数据存储节点上。例如，当 n = 400 时，60%~70%的能耗都是由数据访问操作引起的，而且 IEC 指标中的绝大部分能耗都是由处理器空闲所导致的。原因在于，WIEN2K 的计算工作量相对数据访问工作量而言很小，因此计算节点的空闲时间明显大于数据存储节点。

在任务规模较小时（n =100），实验结果显示，5 种算法的总能耗差别不大，最差的是 HEFT 算法，最优的是 HEFT-MECP。随着 n 的增大，总能耗及其分布都出现了明显差异。值得一提的是，各次实验中的 IEC 指标变化都很小。为了分析任务规模与 DAEC 指标的关系，统计了数据存储节点的资源有效利用率，图 6 为 n=400 时的统计结果。其中，S1 和 S2 为 RAID 节点，S3~S9 为 iSCSI Sever 节点。

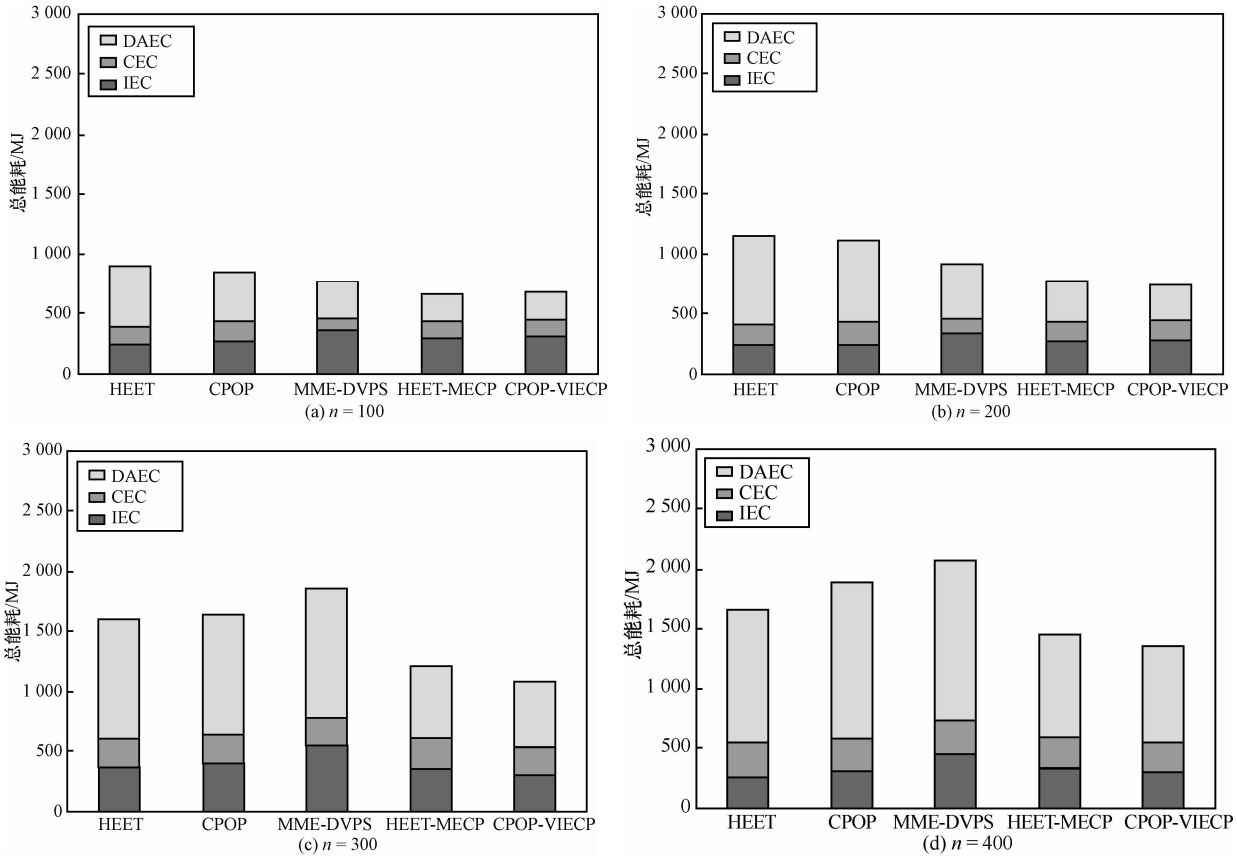


图 5 不同任务规模下的总能耗及其分布

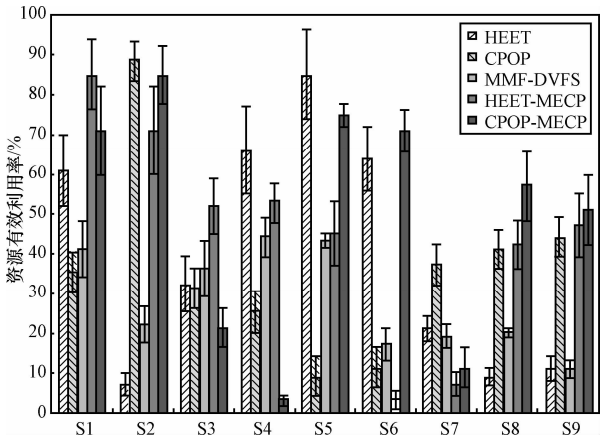


图 6　数据存储节点的资源有效利用率（$n = 400$）

图 6 的统计结果显示，采用 MMF-DVFS 和 CPOP 算法时，各个数据存储节点的利用率比较均衡；而其他 3 种算法则出现了明显的不均衡。HEFT-MECP 和 CPOP-MECP 算法的一个共同特点是，数据存储节点 S1 和 S2 的利用率都很高。在本次实验环境中，S1 和 S2 与计算集群 Crusoe 之间是通过吉比特网络互联，这就直接导致 HEFT-MECP 和 CPOP-MECP 算法在进行调度时偏向于将计算任务调度到 Crusoe 节点，而将数据访问操作映射到 S1 和 S2。

综合以上实验分析，得到如下结论：1) HEFT 和 CPOP 算法能获得较优的调度长度，但当 DAG 任务的数据访问量增大时，其优势会逐渐消失；2) MMF-DVFS 只适用于计算密集型任务的调度；3) HEFT-MECP 和 CPOP-MECP 算法在调度大规模数据密集型任务时充分考虑了带宽因素和节点功耗特性，其能耗优化效果要显著优于 MMF-DVFS 算法以及传统的 HEFT 和 CPOP 算法。

6　结束语

本文针对数据密集型工作流的能耗问题，提出了一种最小能耗路径的启发式策略，并在 HEFT 算法和 CPOP 算法基础实现了 2 种具有能耗感知能力的调度算法（HEFT-MECP 和 CPOP-MECP）。实验结果显示，在面对大型数据密集型工作流时，本文所提算法的调度长度与 HEFT 和 CPOP 算法相当，其数据访问能耗开销则能获得显著减低。相对采用 DVFS 技术的 MMF-DVFS 算法而言，本文所提算法在调度长度和任务执行能耗方面都明显较优。在今后的工作中，将在其他经典 DAG 调度算法（如分层调度算法和聚类调度算法）中引入最小能耗路径的启发式策略，并对其性能表现进行综合测评与分析。此外，还将考虑“冗余调度”策略对系统能耗的影响，并将本文所提的启发策略引入到已有的“冗余调度”算法中。

参考文献：

[1] 胡定磊, 陈书明. 低功耗编译技术综述[J]. 电子学报, 2005, 33 (4): 676-682.
HU D L, CHEN S M. Low power/energy compilation technology[J]. Acta Electronica Sinica, 2005, 33 (4): 676-682.

[2] KANT K, MURUGAN M, DU D C. Willow: a control system for energy and thermal adaptive computing[A]. Proceedings of IEEE International Parallel and Distributed Processing Symposium[C]. Washington, USA, 2011.36-47.

[3] 林闯, 田源, 姚敏. 绿色网络和绿色评价: 节能机制、模型和评价[J]. 计算机学报, 2011, 34(4): 593-612.
LIN C, TIAN Y, YAO M. Green network and green evaluation: mechanism, modeling and evaluation[J]. Chinese Journal of Computers, 2011, 34(4):593-612.

[4] HIKITA J, HIRANO A, NAKASHIMA H. Saving 200kW and $200 K/year by power-aware job/machine scheduling[A]. Proceedings of International Parallel and Distributed Processing Symposium[C]. Washington, USA, 2008.1-8.

[5] KONDO M, IKEDA Y, NAKAMURA H. High performance cluster system design by adaptive power control[A]. Proceedings of International Parallel and Distributed Processing Symposium[C]. Washington, USA, 2007.1-8.

[6] WANG L, LASZEWSKI G, DAYAL J, *et al*. Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS[A]. Proceedings of IEEE/ACM International Conference on Cluster, Cloud and Grid Computing[C]. Washington, USA, 2010.368-377.

[7] RIZVANDI N B, TAHERI J, ZOMAYA A Y, *et al*. Linear combinations of DVFS-enabled processor frequencies to modify the energy-aware scheduling algorithms[A]. Proceedings of IEEE/ACM International Conference on Cluster, Cloud and Grid Computing[C]. Washington, USA, 2010.388-397.

[8] GARG R, SON S W, KANDEMIR M, *et al*. Markov model based disk power management for data intensive workloads[A]. Proceedings of IEEE/ACM International Symposium on Cluster Computing and the Grid[C]. Washington, USA, 2009.76-83.

[9] HU F P, EVANS J J. Power and environment aware control of beowulf clusters[J]. Cluster Computing, 2009, 12(3):299-308.

[10] SONG S, SU C Y, GE R, *et al*. Iso-energy-efficiency: an approach to power-constrained parallel computation[A]. Proceedings of International Parallel and Distributed Processing Symposium[C]. Washington, USA, 2011.128-139.

[11] CHO S, MELHEM R G. On the interplay of parallelization, program performance, and energy consumption[J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 21(3):342-353.

[12] BENOIT A, GOUD P R, ROBERT Y. Performance and energy optimization of concurrent pipelined applications[A]. Proceedings of In-

ternational Parallel and Distributed Processing Symposium[C]. Washington, USA, 2010.1-12.

[13] SHANG P, WANG J. A novel power management for cmp systems in data-intensive environment[A]. Proceedings of International Parallel and Distributed Processing Symposium[C]. Washington, USA, 2011.92-103.

[14] HERATH C, PLALE B. Streamflow-programming model for data streaming in scientific workflows[A]. Proceedings of IEEE/ACM International Conference on Cluster, Cloud and Grid Computing[C]. Washington, USA, 2010.302-311.

[15] KHARGHARIA B, HARIRI S, SZIDAROVSZKY F, *et al*. Autonomic power and performance management for large-scale data centers[A]. Proceedings of International Parallel and Distributed Processing Symposium[C]. Washington, USA, 2007.1-8.

[16] 王桂彬, 杨学军, 徐新海等. 异构系统功耗感知的并行循环调度方法[J]. 软件学报, 2011, 22(9):2222-2234.
WANG G B, YANG X J, XU X B, *et al*. Power-aware parallel loop scheduling method for heterogeneous system[J]. Journal of Software, 2011,22(9):2222-2234.

[17] NIYATO D, CHAISIRI S, SUNG L B. Optimal power management for server farm to support green computing[A]. Proceedings of IEEE/ACM International Symposium on Cluster Computing and the Grid[C]. Washington, USA, 2009.84-91.

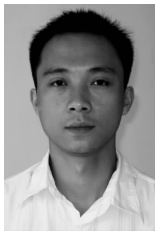
[18] LEE Y C, ZOMAYA A Y. Energy conscious scheduling for distributed computing systems under different operating conditions[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(8): 1374-1381.

[19] TOPCUOGLU H, HARIRI S, WU M Y. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(2):260-274.

[20] CALHEIROS R N, RANJAN R, BELOGLAZOV A, *et al*. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience, 2011, 41(1):23-50.

[21] BLAHA P, SCHWARZ K, MADSEN G, *et al*. WIEN2K: An Augmented Plane Wave Plus Local Orbitals Program for Calculating Crystal Properties[M]. Vienna: Institute of Physical and Theoretical Chemistry, 2001.

作者简介：



肖鹏（1979-），男，湖南湘潭人，博士，湖南工程学院讲师，主要研究方向为高性能网格计算和可信计算。

胡志刚（1963-），男，山西孝义人，中南大学教授、博士生导师，主要研究方向为分布式系统和嵌入式系统。

屈喜龙（1978-），男，湖南新邵人，博士，湖南工程学院副教授，主要研究方向为服务计算和网络集成系统。