DOI：10.7652/xjtuxb201502003

MapReduce 环境中的性能特征能耗估计方法

樊源泉，伍卫国，许云龙，高颜
（西安交通大学电子与信息工程学院，710049，西安）

摘要：针对 MapReduce 系统中负载能耗特征多样性为系统成本调度带来的负载与节点难以匹配的问题，提出一种基于负载性能特征的能耗估计方法。该方法以 MapReduce 系统中各节点操作系统的性能事件为依据估计在线负载的能耗。为了提升负载能耗估计结果的准确度，采用机器学习的方法，在负载执行时，搜集系统的性能特征，并建立估计模型的样本集；采用粗糙集理论中属性约简方法对性能特征属性进行约简；在性能属性约简的结果之上，基于支持向量机理论，建立能耗的估计模型，对负载运行时系统的能耗进行准确的估计。实验结果表明：基于性能特征的能耗估计方法拥有较高的估计准确率，在单作业环境中平均相对误差为 4%，在多作业环境中可达到 4.5%。

关键词：能耗估计；性能特征；MapReduce
中图分类号：TP333 **文献标志码**：A **文章编号**：0253-987X(2015)02-0014-06

A Power Estimation Method Based on Performance Features in MapReduce Environments

FAN Yuanquan, WU Weiguo, XU Yunlong, GAO Yan
(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: It is difficult to improve the energy efficiency of MapReduce clusters by matching active nodes to the needs of the workload since it is difficult to capture the features of energy consumption for cost-based scheduler for different types of workloads. A power estimation method based on performance features of workloads is proposed to solve the problem. The method estimates the power consumption by leveraging performance monitoring counters on components of worker nodes during MapReduce jobs execution. A machine learning method is used to improve the estimation accuracy. The performance monitoring counters of MapReduce system are collected to build a sample set, and then the rough set method is used to select the performance attributes that show strong impact on the energy consumption of workloads. A power estimation model based on the least square support vector machines is built from the attribute reduction results. Experimental results show that the energy estimation method accurately forecasts the power consumption of workloads in MapReduce systems. The relative error of accuracy for power prediction is 4% for only one running job and 4.5% for jobs sharing MapReduce clusters.

Keywords: power estimation; performance features; MapReduce

随着 MapReduce[1] 应用的增多及数据处理规模的剧增，基于 MapReduce 计算模型的集群系统的

收稿日期：2014-05-26。 作者简介：樊源泉(1982—)，男，博士生；伍卫国(通信作者)，男，教授，博士生导师。 基金项目：国家自然科学基金资助项目(61202041，91330117)；国家高技术研究发展计划资助项目(2011AA01A204，2012AA01A306)。
网络出版时间：2014-12-11 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20141211.0849.002.html

能耗问题变得日益突出。平台资源协调器 YARN (yet another resource negotiator)[2]是支持 MapReduce 计算模型的新一代开源系统，提升 YARN 系统中作业执行的能效对于节约成本、保护环境意义重大。但是，YARN 集群中任务随机达到的特性，造成集群中部分工作节点在某些时间段出现空闲等待，系统能源的利用率较低[3]。此外，集群工作节点的异构特性导致系统负载失衡，这也对系统的能效产生负面影响。当前提升 YARN 集群系统能效的方法分为两类：基于工作节点伸缩的节能方法和基于负载成本调度的节能方法[4]。在以上两种方法中，一个核心的问题是负载能耗特征的获取，据此为各个工作节点匹配合适的负载。因此，负载能耗的准确估计是提升 YARN 系统能效的关键。但是，YARN 系统中众多的配置参数、工作节点的异构特性及负载性能特征的多样性为能耗估计带来了挑战。

当前，针对 MapReduce 系统中负载能耗估计的方法有如下两类。第一类为基于硬件性能计数器的能耗估计方法。该类方法在作业执行过程中，采集工作节点中各个组成部件的性能计数器数据，据此估计工作节点的能耗。Rong 等提出了 eTune 系统[5-6]，该系统依赖于专用的能耗测量设备测量工作节点中各个功能部件的能耗，面对数据中心中大规模的工作节点，可用性不强。另外，该方法适用于 CPU 密集型的负载，针对 I/O 密集型的负载，估计结果的准确性不高。第二类方法采用软件测量的方式估计系统的能耗。Fan 等基于 CPU 利用率建立了工作节点的能耗估计模型[7]。针对异构的 Web 服务器，Taliver 等考虑磁盘利用率及 CPU 利用率对能耗的影响，建立了能耗估计模型[8]。但是，以上两种方法仅考虑工作节点中个别组件产生的能耗，预测结果的准确度不高。Suzanne 等开发了 Mantis 系统[9]估计负载产生的能耗。Mantis 系统仅考虑 CPU 及磁盘产生的能耗，将 CPU 利用率和磁盘利用率作为模型的参数，基于线性回归的方法估计负载的能耗，但是当作业在多处理器环境中运行时，Mantis 估计的误差较大。

针对以上问题，本文提出了一种基于负载性能特征的能耗估计方法，该方法能够准确估计MapReduce 系统中作业运行时产生的能耗。

1　能耗估计方法

能耗估计的流程如图 1 所示，其中包括 3 个步骤：负载性能特征数据的离散化、基于粗糙集的性能属性的约简和基于最小平方支持向量机的能耗估计。以下是估计过程的具体描述。

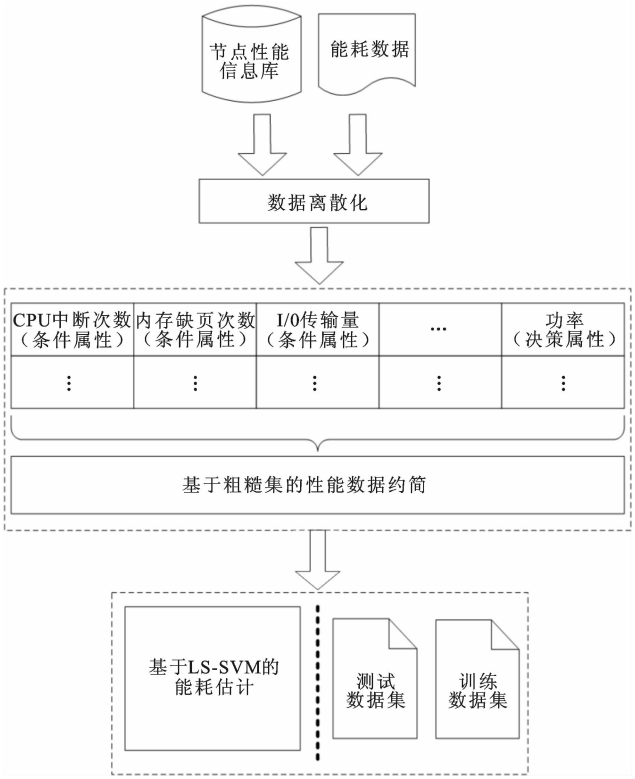


图 1　能耗估计流程

1.1　性能特征数据离散化

在 MapReduce 作业执行过程中，从每个节点采集到的负载性能数据为数值型，首先采用粗糙集理论中的有监督的离散化方法[10-11]对性能数据离散化处理。设 $X=(x_1,x_2,x_3,\cdots,x_n)$ 为离散化处理之前的样本数据集，x_i 由 m 个负载运行时系统的性能事件组成，即 $x_i=(x_{i1},x_{i2},x_{i3},\cdots,x_{im})$，其中 x_{ij} 为任一性能事件。给定任一性能事件 A_i，设阈值为 T，X 被 T 分割成两个子集合，则称 T 为 X 的一个断点。

设离散化处理之前的样本数据集 X 被断点划分为 t 个区间，X 中的任一样本 x_i 所对应的系统实时能耗值为 $P(x_i)$，X 中各个样本对应的不同的系统实时能耗值的个数为 k，X 所包含的总样本个数为 N，第 i 个区间中各样本对应的能耗等于 p_i 的样本总数为 s_{ij}，X 中各样本对应的能耗值为 p_i 的样本总数为 s_i，第 j 个划分区间中样本数为 n_j，则 X 相对于 t 的列联系数为

$$\chi^2(k)=N\left(-1+\sum_{i=1}^{k}\sum_{j=1}^{t}s_{ij}^2/s_in_j\right) \tag{1}$$

设离散化处理之前的样本数据集 X 被划分为 t 个区间，X 中的样本 x_i 所对应的系统实时能耗值为

$P(x_i)$，X 的 t 划分的列联系数为 $\chi^2(t)$，则 X 对 t 划分的评判标准为

$$A(t)=\chi^2(t)/t(k-1) \tag{2}$$

性能数据离散化的过程为求断点集 T，依据 T 中各个断点对离散化处理之前的样本数据集 X 进行划分，在每次划分时使用 $A(t)$ 作为评判，寻找最佳断点，依据最佳断点对每个属性的值域进行划分。

1.2 性能属性的约简

MapReduce 负载的各性能特征属性对能耗的影响不同，为了去除性能特征属性中的冗余，基于粗糙集理论中的信息熵[12]对性能属性约简。设论域 U，P 和 Q 分别为 U 上的两个集合，X 和 Y 分别为由 P 和 Q 确定的 U 的两个划分，其中 $X=(X_1,X_2,X_3,\cdots,X_n)$，$Y=(Y_1,Y_2,Y_3,\cdots,Y_m)$，从操作系统中采集到的负载性能属性的集合为 C，负载运行时各个性能属性采集时间点对应的系统能耗值为 D，X 中的子集 X_i 中所包含的样本的总个数与 U 中所包含的总样本数的比值为 $P(X_i)$，X_i 与 Y 的子集 Y_j 的交集所包含的样本总数与 U 中总样本数的比值为 $P(Y_j|X_i)$。则 U 中各性能属性相对于负载运行时系统产生的能耗的条件信息熵可表示为

$$H(D\mid C)=-\sum_{i=1}^{n}P(X_i)\sum_{j=1}^{m}P(Y_j\mid X_i)\log P(Y_j\mid X_i) \tag{3}$$

定义 1 条件属性集 C 中的属性 C_i 相对于论域 U 的分类重要度表示从 C 中删除 C_i 前后，C 相对于决策属性的信息熵的变化，则属性 C_i 的分类重要度为

$$M(C_i)=H(D\mid C-\{C_i\})-H(D\mid C) \tag{4}$$

对 MapReduce 系统中负载性能属性约简的过程如下：首先，依据离散化之后的条件属性集合 C 及能耗值所对应的决策属性集合 D，依据式(3)计算 D 相对于 C 的信息熵 $H(D|C)$ 及删除某个属性 C_i 之后的信息熵 $H(D|C-\{C_i\})$；其次，针对 C 中每个属性，依据式(4)计算其重要度；最后，依据 C 中各个属性的分类重要度大小，依次删除各个属性 C_i，判断从 C 中删除 C_i 前后，集合 C 相对于集合 D 是否变化，如没有发生变化则保留属性 C_i，否则删除属性 C_i。

1.3 基于最小平方支持向量机的能耗估计模型

针对约简后的性能属性，采用最小平方支持向量机理论[13-14]建立 MapReduce 系统中负载能耗的估计模型。

设 I 为约简之后的负载性能属性，$I=(A_1,A_2,A_3,\cdots,A_m)$，$A_i$ 为负载性能属性集合中第 i 个属性的特征值，则基于最小平方支持向量机的估计模型可表示为

$$P(I)=\sum_{i=1}^{n}\alpha_i K(I,I_i)+a \tag{5}$$

式中：$K(I,I_i)$ 为径向基核函数，可表示为

$$K(I,I_i)=\exp(-\|I-I_i\|^2/\rho^2) \tag{6}$$

式中：ρ 为径向基核函数参数。

能耗估计方法包括以下步骤：先依据约简后的性能属性构建数据集 I，再将数据集 I 切分为训练样本集与测试样本集，采用训练样本集对估计模型进行训练，构建能耗估计模型。

基于最小平方支持向量机的能耗估计算法(MPower)表示如下。

算法 1 MPower 算法

输入 MapReduce 系统中负载性能特征属性 X 与系统产生的能耗值的集合 D，能耗值离散化区间数 k，误差量的最小值 $a_{\min}$ 和最大值 $a_{\max}$，惩罚因子的最小值 $a_{\min}$ 和最大值 $a_{\max}$

输出 能耗估计模型 P

```
/* 决策属性 D 的离散化 */
D'=disConAtt(D)
/* 负载性能特征属性 X 的离散化 */
for each attribute X_i ∈ X do
    x_m ←max(X_i)
    x_o ←min(X_i)
    B_i ←initialBoundary(X_i, x_m, x_o)
    for each b_j ∈ B_I do
      calculate the statistic A(b_j) according to
       equation (2)
      BO_i ←findBestBoundary(A(b_j))
    end for
    BOUND←BO_i
end for
C'=disDecAtt(C,BOUND)
/* 负载性能特征属性约简 */
calculate the Information Entropy of conditional
attribute H(D'|C') according to equation (3)
for each attribute C_i' ∈ C' do
    calculate the importance of conditional attrib-
ute H(D'|C'-{C_i'}) according to equation (3)
    Q← H(D'|C'-{C_i'})
    Ascending(Q)
end for
j=0;
```

$I \leftarrow C'$
while($j<$length(Q)) do
　　if $H(D'|C')=H(D'|I-\{Q_j\})$ do
　　$I \leftarrow I-\{Q_j\}$
　　end if
end while
/ * 基于最小平方支持向量机的能耗估计 * /
$[I^{\text{train}}, I^{\text{test}}] \leftarrow$divide($I$)
for $a \in [a_{\min}, a_{\max}]$ and $\gamma \in [\gamma_{\min}, \gamma_{\max}]$ do
　　$[a_{\text{best}}, \gamma_{\text{best}}] \leftarrow$optimization($a, \gamma, I^{\text{train}}$)
end for

2　实验分析

本节使用普度大学发布的测试用例[15]验证能耗估计方法的有效性，其中包括 WordCount、Sort、Pi、RankedInvertedIndex、RandomWriter 和 TeraGen 等。这些测试用例分别代表 I/O 密集型、CPU 密集型和混合型 3 种 MapReduce 负载类型。

实验的硬件平台为 3 台服务器与 1 台 PC 机构成的 YARN 集群。其中每台服务器的 CPU 为Intel(R) Xeon(R) E5-2420，内存大小为 16 GB，硬盘为 SATA 接口。PC 机的 CPU 为 Intel(R) Core(TM) Q6000 @ 2.40 GHz，内存大小为 2 GB，硬盘为 IDE 接口，存储容量大小为 150 GB。能耗测试仪器为 Watts'up，其额定电压为 250 V，额定电流为 15 A。软件采用的是 Apache 发布的 YARN 平台，版本号为 2.0。

2.1　数据集

本文基于 PerfMon2 实现了性能属性采集器 PProfile。实验中分别向 YARN 集群提交不同类型的作业，并在作业执行过程中对性能特征数据进行采集，采集间隔为 1 s，每次采集 40 个性能属性的特征值。同时，读取采样频率间隔内系统的能耗值，每次采集到的性能数据和能耗值构成一个样本数据，最终构成估计模型的样本数据集。

2.2　结果分析

2.2.1　*负载性能特征属性的约简*　针对浮点型的能耗值，使用等频离散化方法[10]执行离散化操作，属性重要度阈值设为 0.01，得到如表 1 所示的重要度大于 0.01 的性能特征及重要度。

如表 1 所示，YARN 集群系统中负载的能耗与工作节点的各个组成部件均有关联，其中 CPU 对应的性能特征属性的重要度最大，是耗费电能最大的部件。内存所耗费的电能来源于内存的读写，评判内存读写能力的重要指标为每秒钟内存缺页个数和换入换出内存个数，这两个性能数据均在表 1 中有所体现。

表 1　性能特征属性约简结果

性能特征	说明	重要度
CPU_CS	上下文切换次数	0.047 8
CPU_IN	每秒 CPU 中断次数	0.032 0
CPU_UT	用户态时间	0.031 5
CPU_WT	I/O 等待时间	0.021 0
M_PGI	每秒换入内存页面数	0.023 0
M_PGO	每秒换出内存页面数	0.023 0
M_PF	缺页数	0.021 0
N_IN	每秒接收字节数(KB)	0.015 0
N_OUT	每秒发送字节数(KB)	0.015 0
D_W	I/O 等待时间	0.012 2
D_S	I/O 服务时间	0.010 7
D_T	I/O 传输量	0.019 1
D_AS	请求平均扇区数	0.014 8

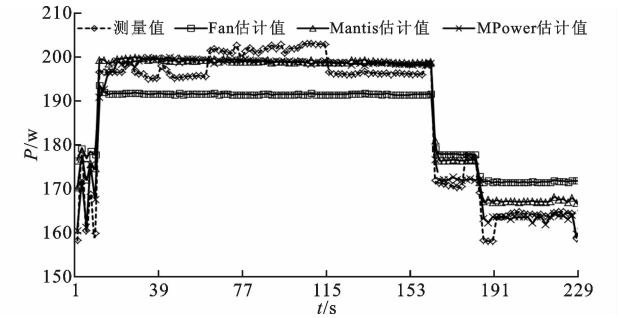
2.2.2　*能耗估计方法的有效性验证*　为了验证 MPower 算法的估计准确度，本小节引入估计相对误差。设 P_i 为 MPower 算法对负载能耗的估计值，Q_i 为测量得到的负载能耗值，则 MPower 算法的估计相对误差可表示为

$$E = | P_i - Q_i | / Q_i \tag{7}$$

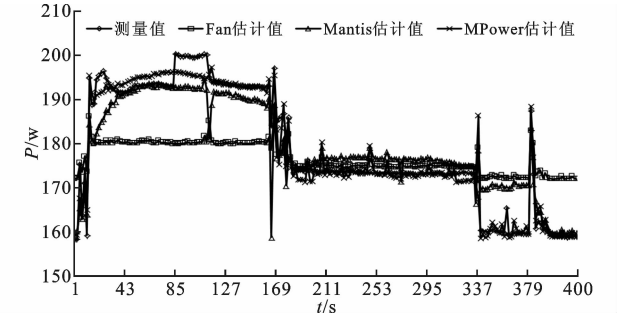
(1)单作业环境。分别向 YARN 集群提交 3 个 Kmeans 作业和 3 个 PI 作业，分别采用 MPower、Mantis[9]与 Fan[7]估计系统能耗，图 2、图 3 为 3 种方法估计结果的比较。

如图 2、图 3 所示，3 种方法均可较为准确地估计负载产生的能耗。对于作业 PI，3 种方法得到了相似的估计结果，这是由于作业 PI 运行过程中系统产生的能耗主要来自于 CPU，而 3 种模型中均考虑了 CPU 的性能特征对能耗的影响。但是，Mantis 与 Fan 方法并没有考虑网络 I/O 对能耗的影响，而这两个影响因素均被 MPower 方法所考虑，因此 MPower 算法的估计准确度较高。对于作业 Kmeans，MPower 的估计准确度明显优于 Mantis 与 Fan 方法，这也是由于 MPower 方法充分考虑了工作节点内部主要组成部件对能耗的影响，而 Fan 方法仅考虑了 CPU 对负载能耗的影响，Mantis 方法仅考虑了磁盘及 CPU 对负载能耗的影响。

(2)多作业环境。在多作业环境中，每种类型的作业拥有不同的特征，因此对集群系统资源的使用



(a)PI



(b)Kmeans

图 2 单作业环境中负载能耗估计

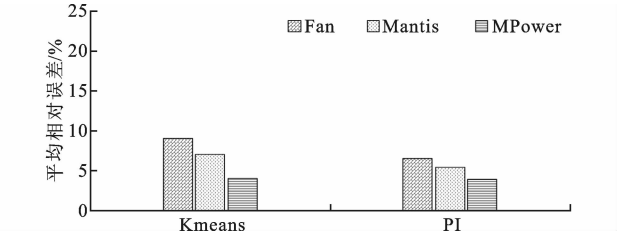


图 3 单作业环境中不同估计模型平均相对误差

表 2 作业类型及输入数据大小

作业组号	作业提交者	作业编号	作业类型	Map 任务个数	Reduce 任务个数	输入数据大小
1	User_1	Job_1	PI	10	1	
	User_2	Job_2	Kmeans	10	3	3 GB
2	User_1	Job_3	PI	10	1	
	User_2	Job_4	Kmeans	10	3	2 GB
3	User_1	Job_5	PI	10	1	
	User_2	Job_6	Kmeans	10	3	1 GB

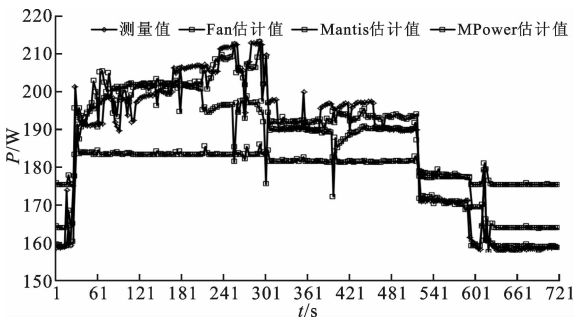


图 4 多作业负载能耗估计

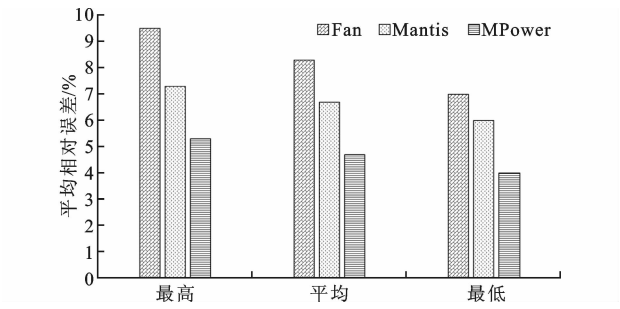


图 5 多作业环境中不同能耗估计方法的平均相对误差

情况也不同。两个用户分别向 YARN 集群提交两种类型的作业 PI 和 Kmeans，如表 2 所示。比较 MPower、Mantis 和 Fan 3 种方法分别在最好、最坏及平均情况下的平均相对误差。图 4、图 5 描述了实验的结果，可以看出 MPower 方法较另外两种方法获得了较高的估计准确率。这主要是因为 MPower 方法中考虑了作业运行时工作节点中主要功能部件产生的能耗，而 Mantis 及 Fan 方法仅考虑了部分功能部件耗费的电能。此外，对比图 2 和图 4，单作业或多作业环境中负载产生的能耗与操作系统中采集到的性能特征并非简单的线性相关，MPower 方法中对性能特征进行属性约简，保留贡献较大的性能特征，以此建立非线性估计模型，对负载的能耗值进行估计，因此获得了较低的估计平均相对误差。

总之，在多作业和单作业环境中，MPower 估计准确度高于已有的 Fan 方法和 Mantis 方法。

2.2.3 *能耗估计方法的应用场景* 在 YARN 集群中，一种常用的节能机制是动态电压可扩展技术(DVFS)。为此，可将本文所提能耗估计方法与 DFVS 技术结合制定节能策略。如图 2a 所示，PI 用例的能耗具有很强的规律性，在 161 s 之前 PI 用例处于 Map 阶段，能耗较高，但 161 s 之后 PI 用例进入 Shuffle 阶段，能耗明显降低，该规律已被 MPower 监控到，此时工作节点可以启动 DVFS 机制降低 CPU 主频，从而降低负载的能耗。在 185 s 之后 PI 用例进入 Reduce 阶段的后期，此时，从 MPower 估计结果可以看出工作节点的能耗进一步降低。此时，可启动 DVFS 技术降低 CPU 能耗从而达到节能目的。

此外，MPower 可用来制定基于成本的调度策略。例如，结合 MPower 估计的能耗值及节点的性能特征，在保证服务质量的前提下，将 Map 或 Reduce 任务调度至当前能耗值较低的节点，而将部分节点休眠或关闭，从而提高 YARN 集群的能效。

3　结　论

本文针对 YARN 集群提出了一种负载能耗的估计方法，该方法基于负载的性能特征估计工作节点的能耗。针对 Fan 与 Mantis 等线性模型估计结果的平均相对误差高的问题，本文依据 MapReduce 作业运行时的系统性能特征，利用粗糙集理论中的属性约简方法来约简性能特征属性，并基于属性约简的结果来构建能耗估计模型。实验结果表明：本文所提方法估计结果的平均相对误差为 4.5%，低于已有的 Fan 和 Mantis 方法。

参考文献：

[1] DEAN, JEFFREY, SANJAY G. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 1(1): 107-113.

[2] VAVILAPALLI V K, MURTHY A C, DOUGLAS C, et al. Apache Hadoop YARN: yet another resource negotiator [C]// Proceedings of the 4th Annual Symposium on Cloud Computing. New York, USA: ACM, 2013: 1-16.

[3] 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.

[4] LEVERICH, JACOB, CHRISTOS K. On the energy (in) efficiency of hadoop clusters [J]. ACM SIGOPS Operating Systems Review, 2010, 44(1): 61-65.

[5] GE Rong, FENG Xizhou, WIRTZ T, et al. eTune: a power analysis framework for data-intensive computing [C]// Proceedings of the 2012 41st International Conference on Parallel Processing Workshops. Piscataway, NJ, USA: IEEE, 2012: 254-261.

[6] WIRTZ T, GE Rong. Improving Mapreduce energy efficiency for computation intensive workloads [C]// Proceedings of the 2011 International Green Computing Conference and Workshops. Piscataway, NJ, USA: IEEE, 2011: 1-8.

[7] FAN Xiaobo, WEBER W D, BARROSO L A. Power provisioning for a warehouse-sized computer [J]. ACM SIGARCH Computer Architecture News, 2007, 35(2): 13-23.

[8] HEATH T, DINIZ B, CARRERA E V, et al. Energy conservation in heterogeneous server clusters [C]// Proceedings of the 10th ACM Sigplan Symposium on Principles and Practice of Parallel Programming. New York, USA: ACM, 2005: 186-195.

[9] RIVOIRE S, RANGANATHAN P, KOZYRAKIS C. A comparison of high-level full-system power models [J]. HotPower, 2008, 8: 1-5.

[10] LI Bing, CHOW T W S, TANG Peng. Analyzing rough set based attribute reductions by extension rule [J]. Neurocomputing, 2014, 123: 185-196.

[11] HAN Jiawei, KAMBER M, PEI Jian. Data mining: concepts and techniques [M]. San Francisco, CA, USA: Morgan Kaufmann, 2006: 63-70.

[12] 王国胤，姚一豫，于洪. 粗糙集理论与应用研究综述 [J]. 计算机学报，2009，32(7)：1229-1246.
WANG Guoyin, YAO Yiyu, YU Hong. A survey on rough set theory and applications [J]. Journal of Computers, 2009, 32(7): 1209-1246.

[13] CORTES C, VAPNIK V. Support vector machine [J]. Machine Learning, 1995, 20(3): 273-297.

[14] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999, 9(3): 293-300.

[15] AHMAD F, CHAKRADHAR S T, RAGHUNATHAN A, et al. Tarazu: optimizing Mapreduce on heterogeneous clusters [C]// Proceedings of the 17th International Conference on Architectural Support for Programming Languages and Operating Systems. New York, USA: ACM, 2012: 61-74.

[本刊相关文献链接]

袁通，刘志镜，刘慧，等. 多核处理器中基于 MapReduce 的哈希划分优化. 2014，48(11)：97-102. [doi：10.7652/xjtuxb201411017]

史槐，耿晨，齐勇. 一种具有容错机制的 MapReduce 模型研究与实现. 2014，48(2)：1-7. [doi：10.7652/xjtuxb201402001]

崔华力，钱德沛，张兴军，等. 用于无线多跳网络视频流传输的优先级机会网络编码. 2013，47(12)：13-18. [doi：10.7652/xjtuxb201312003]

陈衡，钱德沛，伍卫国，等. 传感器网络基于邻居信息量化的能量平衡路由. 2012，46(4)：1-6. [doi：10.7652/xjtuxb201204001]

（编辑　武红江）