# 基于作业历史运行信息的MapReduce能耗预测模型

廖 彬[1] 张 陶[2] 于 炯[3] 孙 华[3]
(新疆财经大学统计与信息学院 乌鲁木齐 830012)[1] (新疆医科大学医学工程技术学院 乌鲁木齐 830011)[2]
(新疆大学软件学院 乌鲁木齐 830008)[3]

**摘 要** 在数据量规模剧增的背景下,大数据处理过程中产生的高能耗问题亟待解决,而能耗模型是研究提高能耗效率方法的基础。利用传统的能耗模型计算MapReduce作业执行能耗面临诸多挑战,在对大数据计算模型MapReduce的集群结构、作业的任务分解及任务与资源映射模型分析建模的基础上,提出基于作业历史运行信息的MapReduce能耗预测模型。通过对不同作业历史运行信息的分析,得到DataNode运行不同任务时的计算能力及能耗特性,继而实现在MapReduce作业执行前对作业能耗的预测。实验结果验证了能耗预测模型的可行性,并通过对能耗预测准确率调节因子的修正,能够达到提高能耗模型的预测准确度的目的。

**关键词** 绿色计算,MapReduce,能耗建模,预测模型
**中图法分类号** TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.037

## Prediction Model of Energy Consumption for MapReduce Based on Job Running History Logs

LIAO Bin[1] ZHANG Tao[2] YU Jiong[3] SUN Hua[3]
(College of Statistics and Information, Xinjiang University of Finance and Economics, Urumqi 830012, China)[1]
(Department of Medical Engineering and Technology, Xinjiang Medical University, Urumqi 830011, China)[2]
(School of Software Design, Xinjiang University, Urumqi 830008, China)[3]

**Abstract** The problem of high energy consumption produced in big data processing is an important issue that needs to be solved, especially under the background of data explosion. The energy consumption model is the basis for research to improve the energy efficiency of the MapReduce. Using traditional model to calculate the MapReduce job's energy consumption faces challenges. After research on the cluster structure, job task decomposition and task slot mapping mechanism, we proposed the prediction model of energy consumption for MapReduce based on job running history logs. Through the analysis of historical operating information of different jobs, we got the computing power and energy consumption characteristics of DataNode running different tasks, and then implemented the forecast of the energy consumption of the MapReduce job before its execution process. The experimental results demonstrate the feasibility of energy prediction model, and the purpose of improving the prediction accuracy of the model can be achieved by adjusting the correction factor.

**Keywords** Green computing, MapReduce, Energy consumption modeling, Prediction model

## 1 引言

随着云计算、物联网、移动互联网等技术的不断发展,数据正以前所未有的速度在不断增长和积累。数据的产生过程在经历被动和主动两种产生过程后发展到了自动产生阶段,这预示着大数据时代的来临。数据从简单的处理对象开始转变为一种基础性资源,如何更好地管理和利用大数据已经成为普遍关注的话题,大数据的规模效应给数据存储、管理以及数据分析带来了极大的挑战[1]。据文献[2]统计,2007年全球数据总量达到了281EB,并且从2007年到2011年这短短5年时间内,全球数据量增长了10倍,数据以每年平均2倍的速度快速增长。与数据量的高速增长伴随而来的是存储与处理系统规模不断扩大,这使得运营成本不断提高,其成本不仅包括硬件、机房、冷却设备等固定成本,还包括IT设备与冷却设备的电能消耗等其它开销;并且,系统的高能耗将导致过量温室气体的排放并引发环境问题。事实上,在能源价格上涨、数据中心存储规模不断扩大的今天,高能耗已逐渐成为制约云计算与大数据快速发展的一个主要瓶颈[1]。据文献[3]统计,目前由IT领域所产生的二氧化碳排放量占全球总排放量的2%,而到2020年将增长到4%。2008年全球路由器、交换

到稿日期:2014-05-18 返修日期:2014-07-28 本文受国家自然科学基金项目:多Slot环境下的MapReduce能耗模型及优化研究,新疆财经大学博士启动基金项目:大数据存储层节能模型及算法研究资助。

廖 彬(1986—),男,博士,讲师,CCF会员,主要研究方向为数据库技术、云计算与绿色计算,E-mail:liaobin665@163.com;张 陶(1988—),女,硕士,主要研究方向为分布式计算、网格计算;于 炯(1964—),男,博士,教授,博士生导师,主要研究方向为网络安全、网格与分布式计算;孙 华(1977—),女,博士,副教授,主要研究方向为网络安全与云计算。

机、服务器、冷却设备、数据中心等互联网设备总共消耗 8680 亿度电，占全球总耗电量的 5.3%。纽约时报与麦肯锡经过一年的联合调查，最终在《纽约时报》上发表了“Power, pollution and the Internet”[4]，调查显示 Goolge 数据中心年耗电量约 300 万瓦，Facebook 则达到了 60 万瓦，但巨大的能耗中却只有 6%～12%的能耗被用于处理相应用户的请求。

MapReduce[5]作为当前最流行的大数据计算模式，同样面临着严重的能耗问题。以 MapReduce 的提出者 Google 为例，文献[6]中对 Google 内部 5000 多台服务器进行长达半年的调查统计结果表明：服务器在大部分时间里利用率都在 10%～50%之间，服务器在负载很低（小于 10%）的情况下电能消耗也超过了峰值能耗的 50%。MapReduce 的开源实现 Hadoop 由于能够部署在通用平台上，并且具有可扩展性（scalable）、低成本（economical）、高效性（efficient）与可靠性（reliable）等优点，在分布式计算领域得到了广泛运用，并且已逐渐成为工业与学术届事实上的海量数据并行处理标准[7]。虽然 Hadoop 拥有诸多优点，但是与 Google 服务器一样，Hadoop 集群内部服务器同样存在严重的高能耗低利用率问题[8]。Hadoop 主要由分布式存储系统 HDFS 与分布式任务执行框架 MapReduce 两部分组成，现有研究大多从分布式存储系统 HDFS 入手解决 Hadoop 的能耗问题，针对 MapReduce 计算框架能耗问题的研究则相对较少。本文在对MapReduce 的集群结构、作业的任务分解及任务与资源映射模型分析建模的基础上，提出基于作业历史运行信息的MapReduce能耗预测模型，实现在 MapReduce 作业执行前对作业能耗进行预测，为节能的 MapReduce 任务调度研究打下基础。

本文第 2 节对相关工作进行了介绍；第 3 节提出了基于作业历史运行信息的 MapReduce 能耗预测模型；第 4 节通过实验对本文提出的能耗预测模型进行了验证；最后对全文进行了总结。

## 2 相关工作

学术与工业界分别从硬件[9-11]、操作系统[12]、虚拟机[13-17]、数据中心[18-20]4 个层次去解决 IT 系统的能耗问题。针对分布式计算系统（如 MapReduce）的能耗问题的研究，通常以 Hadoop 作为研究对象，并且大多从分布式存储系统 HDFS 入手解决其存在的能耗问题[21-24]，针对 MapReduce 能耗问题的研究则相对较少。可将分布存储系统中的能耗优化问题划分为基于硬件的节能与基于调度的节能两个方面。基于硬件的节能方法主要通过低能耗高效率的硬件设备或体系结构对现有的高能耗存储设备进行替换，来达到节能的目的。这种方法的效果立竿见影，且不需要复杂的能耗管理组件；但是对于已经部署的大规模应用系统，大批量的硬件替换面临成本过高的问题。基于调度的节能通过对存储资源的有效调度，在不影响系统性能的前提条件下将部分存储节点调整到低能耗模式（如休眠、降频等），达到节能的目的。

在 MapReduce 能耗问题方面，已有研究通过选择部分节点执行任务[25]、任务完成后关闭节点[26]、配置参数优化[27]、DVFS 调度[28]、作业调度[29]、虚拟机放置策略[30]及数据压缩[31]等方法达到提高 MapReduce 能耗利用率的目的。文献[26]提出 All-In Strategy（AIS）策略，即将整个 MapReduce 集群作为整体用于任务的执行，当任务结束后将整个集群做节能处理（关闭节点）来达到节能的目的。Chen Y 等人[25]发现 MapReduce框架的参数配置对 MapReduce 能耗的利用具有较大影响，通过大量的实验得到优化 MapReduce 能耗的配置参数，对提高 MapReduce 集群系统的能耗利用率具有指导意义。文献[28]利用 DVFS（Dynamic Voltage and Frequency Scaling）技术，通过动态地调整 CPU 频率以适应当前的 MapReduce任务负载状态达到优化能耗利用的目的。文献[29]提出 Hadoop 节能适应性框架 GreenHadoop，其通过合理的作业调度，在满足作业截止时间约束的前提下通过配置与当前作业量相匹配的作业处理能力（活动节点数量），达到最小化 Hadoop 集群能耗的目的，实验证明 GreenHadoop 与 Hadoop 相比提高了 MapReduce 的能耗利用率。宋杰等人[32]对云数据管理系统（包括基于 MapReduce 的系统）的能耗进行了基准测试，证明了不同系统在能耗方面存在着较大差异，需要进一步对系统进行能耗优化。以上工作主要是研究提高 MapReduce 能耗的方法，而本文则是研究 MapReduce 能耗预测模型，实现在 MapReduce 作业执行前对作业能耗进行预测，为节能的 MapReduce 任务调度研究打下基础。本文主要做了如下几个方面的工作：

（1）对大数据计算模式 MapReduce 的系统模型进行了建模，包括系统集群结构模型、作业的任务分解模型及任务与资源映射模型 3 个方面。

（2）分析了利用传统的能耗模型计算 MapReduce 作业执行能耗面临的问题，通过对不同作业历史运行信息的分析，得到 DataNode 节点运行不同任务时的计算能力及能耗特性，继而实现在 MapReduce 作业执行前对作业能耗进行预测。

（3）通过大量的实验及能耗数据分析，表明本文所提能耗预测模型的有效性，并且通过对能耗预测准确率调节因子的不断修正，提高能耗模型的预测准确度，使得能耗预测模型更加可行。

## 3 MapReduce 能耗预测模型

3.1 节对 MapReduce 系统模型进行了建模，包括集群模型、作业任务分解模型及任务资源映射模型；3.2 节对传统的 MapReduce 任务能耗模型进行了介绍；3.3 节提出了 MapReduce 能耗预测模型。

### 3.1 MapReduce 系统模型

MapReduce 系统通常由多个机架（RACK）组成，而一个 RACK 内部又由多个节点服务器组成。通常情况下，MapReudce集群由两个 NameNode 管理节点（主管理节点与从管理节点）与多个 DataNode 节点构成。在实际应用环境中，考虑到 DataNode 节点数量远远大于 NameNode 节点，本文的能耗模型主要针对 DataNode 节点，不对 NameNode 节点进行能耗建模。

**定义 1**（MapReudce 集群 DataNode 节点模型） 将MapReudce集群中 $n$ 个 DataNode 节点用集合 $DN$ 表示：

$$DN=\{dn_0,dn_1,\cdots,dn_{n-1}\} \tag{1}$$

其中，$dn_i \in DN(0\leqslant i\leqslant n-1)$ 表示 MapReudce 集群中的 DataNode节点。

**定义 2**（作业的任务分解模型） 如图 1 所示，MapReduce

框架将作业(Job)分解为多个 Map 与 Reduce 任务并行地在集群中执行，可将这个过程定义为 $f(Job)\mapsto M\cup R$，其中 $f$ 为映射函数，集合 $M$ 与 $R$ 分别表示 Job 分解后的 Map 与 Reduce 任务，其映射模型可由式(2)表示：

$$\begin{cases} f(Job)\mapsto M\cup R \\ M=\{mt_0,mt_1,\cdots,mt_{m-1}\} \\ R=\{rt_0,rt_1,\cdots,rt_{r-1}\} \end{cases} \tag{2}$$

式(2)表示作业(Job)被分解为 $m$ 个 Map 任务与 $r$ 个 Reduce任务。其中，$mt_i(mt_i\in M,0\leqslant i\leqslant m-1)$表示任意 Map 任务，$rt_i(rt_i\in R,0\leqslant i\leqslant r-1)$表示任意 Reduce 任务。

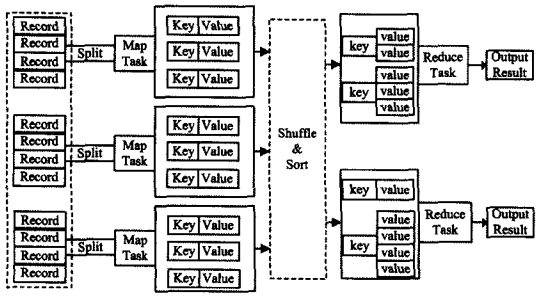


图 1　MapReduce 作业的任务分解

**定义 3**(任务(Task)资源(Slot)映射模型)　作业(Job)被分解为 Map 与 Reduce 任务后，将由作业调度系统将任务映射到具有空闲资源槽(Slot)的 DataNode 节点 $dn_i$ 上执行。任务与资源映射过程可定义为映射 $f(M\cup R)\mapsto DN$，其中 $f$ 为映射函数，$M$ 与 $R$ 分别表示 Map 与 Reduce 任务的集合，$DN$ 表示 MapReudce 集群中 DataNode 节点的集合。具体而言，映射模型可由式(3)具体描述：

$$\begin{cases} f(M\cup R)\mapsto DN \\ M=\{mt_0,mt_1,\cdots,mt_{m-1}\} \\ R=\{rt_0,rt_1,\cdots,rt_{r-1}\} \\ DN=\{dn_0,dn_1,\cdots,dn_{n-1}\} \end{cases} \tag{3}$$

任务资源映射模型由 MapReduce 的任务调度模型决定，现有的主流 MapReduce 调度器有 FIFO、Fair、Capacity、LATE 及 Deadline Constraint 等。

### 3.2　传统的 MapReduce 任务能耗模型

当作业任务调度完毕后，设某任务 $task(task\in M\cup R)$被映射到具有空闲资源槽(Slot)的 DataNode 节点 $dn_i$ 上执行。$task$ 任务从时间点 $t_s$ 开始，运行到时间点 $t_e$ 结束，即作业运行时间区间为$[t_s,t_e]$。那么，运行任务 $task$ 所需要的能耗 $E_{task}$ 可由式(4)得到：

$$E_{task}=\int_{t_s}^{t_e}P_{dn_i}(u_i(t))\mathrm{d}t \tag{4}$$

其中，$u_i(t)$表示在 $t(t\in[t_s,t_e])$时刻节点 $dn_i$ 的系统利用率(也可分别表示 CPU、内存、磁盘、网卡等部件的利用率)，并且存在 $u_i(t)\in[0,1]$；$P_{dn_i}(u_i(t))$则表示在 $t$ 时刻 $u_i(t)$系统利用率条件下 $dn_i$ 节点的功耗。

节点 $dn_i$ 的电能消耗是由 CPU、内存、磁盘、网卡等设备的电能消耗共同组成的。节点瞬时功率值为这些硬件部件的瞬时功率之和。由于 CPU、内存、磁盘、网卡等部件的电器特性之间存在着很大的差异，导致不同部件能耗计算方法不同。使用累加所有部件能耗(或瞬时功耗)来计算节点的能耗值的方法较为复杂。因此，3.3 节提出了 MapReduce 任务能耗预测模型解决传统的 MapReduce 任务能耗模型面临的复杂性问题。

### 3.3　MapReduce 任务能耗预测模型

设 $C_{ij}$ 表示节点 $dn_i$ 处理任务 $task_j$ 时的计算能力，$T_{ij}$ 表示节点 $dn_i$ 完成任务 $task_j$ 所花的时间，$D_{ij}$ 表示任务所处理数据量的大小，在任务执行完毕后，通过任务执行日志可知 $T_{ij}$ 与 $D_{ij}$ 的值，$C_{ij}$ 可由式(5)计算：

$$C_{ij}=\frac{D_{ij}}{T_{ij}} \tag{5}$$

MapReduce 集群中的 DataNode 节点对于不同任务的计算能力可通过大量的任务运行日志信息统计得到，本文中设 $C_{ij}$ 值为已知(见表 1)。可将统计得到的不同节点处理不同任务时的计算能力用表 1 进行记录。

表 1　节点对于不同任务的计算能力

| 任务节点 | $dn_1$ | $dn_2$ | … | $dn_i$ |
|---|---|---|---|---|
| $task_1$ | $C_{11}$ | $C_{21}$ | … | $C_{i1}$ |
| $task_2$ | $C_{12}$ | $C_{22}$ | … | $C_{i2}$ |
| … | … | … | … | … |
| $task_j$ | $C_{1j}$ | $C_{2j}$ | … | $C_{ij}$ |

作业的任务分解及任务(定义 1)与资源之间的映射关系(定义 2)由作业调度系统(如 FIFO、Fair、Capacity、LATE 等)确定。当某任务 $task(task\in M\cup R)$，设数据量大小为 $D_i$)调度到节点 $dn_i$ 上执行时，其运行时间 $T_i$ 可通过式(5)进行预测(节点 $dn_i$ 对任务 $task_i$ 的计算能力可通过大量数据训练获得)。设任务 $task_i$ 在节点 $dn_i$ 上运行的时间区间 $T_i$ 内，节点 $dn_i$ 的平均功耗为$\overline{p_i}$。同样，$\overline{p_i}$的值可通过大量的实验及数据分析获得。当 $task_i$ 的运行时间 $T_i$ 及平均功耗$\overline{p_i}$确定时，$task_i$的能耗可由式(6)估算：

$$E_{task}=\sigma\cdot\overline{p_i}\cdot T_i=\sigma\cdot\overline{p_i}\cdot\frac{D_i}{C_i} \tag{6}$$

其中，$\sigma$ 为能耗预测准确率调节因子，可通过对 $\sigma$ 值的调节对能耗预测准确度进行修正。利用基于平均功耗估算的能耗模型对 MapReduce 作业进行能耗计算的唯一前提条件是取得作业的分解与调度结果。作业的分解与调度结果由 MapReduce 的作业调度系统确定，现有的调度系统在考虑集群 CPU、内存、磁盘及网络等资源状态的基础上，基于作业优先级、截止时间、作业量、作业类型等因素对作业进行调度。现有调度系统并没有将能耗作为一种资源进行考虑，而基于平均功耗估算的能耗模型的主要功能是在作业运行前对作业的执行能耗进行预测，所以基于平均功耗估算的能耗模型是将来研究节能的 MapReduce 作业调度系统的基础。

MapReduce 接收到一个 Job 作业后，首先按照定义 2(作业的任务分解模型)将 Job 分解为多个 Map 与 Reduce 任务；当 Job 进入运行状态后，MapReduce 框架按照定义 3 中的资源映射模型将所有的任务绑定到一个空闲资源槽执行。MapReduce 作业的执行能耗为所有子任务(Map 任务与 Reduce 任务)能耗的总和，设 $E_{Job}$ 表示某作业的执行能耗，基于平均功耗估算的能耗模型 $E_{Job}$ 计算公式推导如式(7)所示：

$$\begin{aligned} E_{Job}&=(E_{map1}+E_{map2}+\cdots+E_{map\_m})+(E_{reduce1}+E_{reduce2}+\\ &\quad\cdots+E_{reduce\_r}) \\ &=\sum_{i=1}^{m}(\sigma_m\cdot\overline{p_i}\cdot\frac{D_i}{C_i})+\sum_{u=1}^{r}(\sigma_r\cdot\overline{p_u}\cdot\frac{D_u}{C_u}) \end{aligned} \tag{7}$$

其中，$\sigma$ 表示能耗预测准确率调节因子，$\overline{p}$表示任务运行过程中的平均功耗，$D$ 表示任务数据量大小，$C$ 表示节点执行该任务时的计算能力。

## 4 实验及结果分析

4.1 节对实验环境及实验的作业类型配置进行说明；4.2 节首先对作业历史运行信息（包括运行时间及能耗）进行记录，通过统计得出不同作业 Map 与 Reduce 任务的平均功耗及计算能力；在 4.2 节的基础上，4.3 节运用能耗模型对作业能耗进行预测，并对误差进行分析，通过调节能耗预测准确率因子对误差进行修正。

### 4.1 实验环境及作业配置

为了对提出的 MapReduce 能耗预测模型进行实验验证，搭建了由 12 个节点组成的 Hadoop 集群；其中 NameNode 与 SecondNameNode 分别独立为 1 个节点，其余 10 节点为 DataNode（设置每个 RACK 中 5 个 DataNode 节点）。为了控制实验过程中的 Map 任务数量，将单个 DataNode 节点上 Map 与 Reduce 任务 Slot 资源槽数设置为 1，即配置项 mapred. tasktracker. map. tasks. maximum 与 mapred. tasktracker. reduce. tasks. maximum 的值都设置为 1。能耗数据测量方面，实验采用北电电力监测仪 USB 智能版，数据采样频率设置为 1 次/秒，各节点能耗数据可通过 USB 接口实时地传输到能耗数据监测机上，实现能耗数据的实时收集。实验总体环境描述如表 2 所列。

表 2 总体实验环境描述

| 项目 | 描述 |
|---|---|
| 操作系统 | Debian 7.5 |
| Java 版本 | 1.6 for Linux |
| Hadoop | 2.1.0 beta |
| 能耗数据测量 | 北电电力监测仪（USB 智能版），标准为 GB/T17215-2003，功率误差值±0.01～0.1W，采样频率为 1.5s～3s 之间，单位为 kWh |
| 能耗数据采集 | 电力监测仪用电监测管理系统 V1.0.1 |
| 能耗相关单位 | 功率：瓦特（W），能耗：焦耳（J） |
| 数据采样频率 | 1 秒采集数据 1 次 |
| 节点 CUP | Intel core2 duo E8400 3.00GHz |
| 节点内存 | 2GB-DDR2-800MHZ |
| 节点硬盘 | Hitachi HDP725032GLA380（320G 7200 转/s） |
| 网卡信息 | Realtek RTL8168/8111 PCI-E Gigabit Ethernet NIC-100Mbps |

实验作业类型方面，本文选取 WordCount、TeraSort、NuthIndex、K-means、Bayes 及 PageRank 6 种 MapReduce 作业进行实验，作业参数设置如表 3 所列。

表 3 作业类型说明

| 名称 | 参数配置 |
|---|---|
| WordCount | 数据总量为 9759.6M，Map 任务数为 10，Reduce 任务数为 10 |
| TeraSort | 数据总量为 9536.7M，Map 任务数为 10，Reduce 任务数为 10 |
| NutchIndex | Page 数量为 1000000，Map 任务数为 80，Reduce 任务数为 10 |
| K-means | Cluster 数为 5，Sample 数为 40000000，每个 Input 文件中的 Sample 数为 4000000，dimensions 大小为 20，最大迭代次数为 1，Map 任务数为 10，Reduce 任务数为 1 |
| Bayes | Page 数目为 50000，分类数目为 100，参数 ngrams＝3，Map 任务数为 10，Reduce 任务数为 1 |
| PageRank | Page 数目为 3000000，迭代次数设置为 3，参数 Block 与 Block_width 分别设置为 0 与 16，Map 任务数为 10，Reduce 任务数为 1 |

### 4.2 构造节点的计算能力表

为了减小实验误差，将 WordCount、TeraSort、NuthIndex、K-means、PageRank 及 Bayes 6 种作业分别执行 10 次取平均数。记录各作业 Map 与 Reduce 任务的开始与结束时间，通过能耗监测仪的采样数据得到每个任务的实时执行能耗。任务的平均功耗由任务的执行能耗除以任务的执行时间得到，节点对任务的计算能力可通过式(5)计算获得。通过对大量实验数据的统计分析，得到 6 种作业 Map 与 Reduce 任务的平均功耗及计算能力情况，如表 4 所列。其中，WordCount 与 TeraSort 单任务处理数据量都为 1024Mb，PageRank单任务处理 300000 个页面，K-means 单任务处理 4000000 个 SAMPLE，Bayes 单任务处理 30000 个随机生成的页面。

需要注意的是，由于 MapReduce 作业分为 Map 与 Reduce 两个阶段，因此必须针对 Map 与 Reduce 两个阶段分别计算任务的平均功耗及节点计算能力。其中，WordCount、TeraSort 与 NuthIndex 不存在阶段性任务，分解较为简单。而 K-means、PageRank 及 Bayes 3 个作业由于存在阶段性任务，需要计算每个阶段的 Map 与 Reduce 任务的平均功耗及节点计算能力，相比 WordCount、TeraSort 与 NuthIndex 3 种作业，复杂度较高。

表 4 作业 Map 与 Reduce 任务的平均功耗及计算能力

| 作业名称 | 阶段性任务 | Map 任务平均功耗 | Map 计算能力 | Reduce 任务平均功耗 | Reduce 计算能力 |
|---|---|---|---|---|---|
| WordCount | N/A | 89.526W | 1.3064M/s | 71.2W | 81.3233M/s |
| TeraSort | N/A | 83.4W | 5.96M/s | 82.75W | 5.36M/s |
| NutchIndex | N/A | 89W | 436.68 page/s | 90.5W | 233.1 page/s |
| K-means | Cluster Iterator | 77.46W | 5797.1sample/s | 77.47W | 7130.1sample/s |
| | Cluster Classification | 81.2W | 7619sample/s | N/A | N/A |
| PageRank | Stage1： | 83.5W | 3333.33page/s | 80W | 1685.393page/s |
| | Stage2： | 84W | 3333.33page/s | 89.42W | 7692.3page/s |
| Bayes | DocumentProcessor：DocumentTokeuizer | 84.7W | 681.8page/s | 79.02W | 1111page/s |
| | CollocDriver. generateCollocations | 80.1W | 12.85page/s | 100.7W | 158.73page/s |
| | CollocDriver. computeNGrams | 80.55W | 434.78page/s | 80.98W | 833.33page/s |
| | DictionaryVectorizer：MakeParitialVectors | 91.59W | 1250page/s | 84.25W | 1111page/s |
| | ParticalVectorMerger：MergePartialVectors | 82.6W | 4761.9page/s | 72.85W | 2439.02page/s |
| | VectorTfld Document Frequency Count | 71.39W | 4966.89page/s | 73.755W | 2481.6page/s |
| | MakePratialVectors | 85.42W | 4862.5page/s | 74.256W | 2479.3page/s |
| | ParticalVectorMerger：MergePartialVectors | 75.92W | 4874.2page/s | 73.26W | 2463.6page/s |
| | TrainNaiveBayesJob IndexInstancesMapper-Reducer | 70.18W | 4951.45page/s | 75.18W | 2015.48page/s |
| | TrainNaiveBayesJob WeightsMapper-Reducer | 77.05W | 4858.04page/s | 71.23W | 2562.82page/s |

### 4.3 能耗模型运用及因子调节

本节将基于4.2节表4中的数据及能耗预测模型公式，对能耗预测模型的可行性进行验证。

在作业运行前根据作业的任务分解及调度结果，运用基于平均功耗估算的能耗模型对6种作业的能耗进行估算，各任务不同阶段的计算能力及平均功耗见表4。作业运行过程中利用能耗监测仪对各任务执行能耗（任务开始到任务结束时间段内节点能耗）进行监测并对数据进行记录，累加得到作业能耗实际值，并将此能耗值作为基准值。此外，由于能耗模型是对能耗进行估算，难免存在一定程度的误差，因此表5对能耗模型的估算值、能耗测量值及估算误差进行了记录。

设$\sigma=1$，以WordCount任务为例，当作业处理数据量为5000MB时，经过任务调度后，每个DataNode节点需要处理500MB的数据，根据式(6)，节点处理Map任务的能耗为：

$$E_{wordcount\_map}=\sigma\cdot\bar{p}\cdot\frac{D}{C}=1\times89.526\times\frac{500}{1.3064}$$
$$=34264.3907\text{J}$$

节点处理Redcue任务的能耗为：

$$E_{wordcount\_reduce}=\sigma\cdot\overline{p_r}\cdot\frac{D_r}{C_r}=1\times71.2\times\frac{500}{81.3233}$$
$$=437.75892\text{J}$$

通过式(7)可得到WordCount作业的能耗为：

$E_{job}=10\times(34264.3907+437.75892)\approx347021.5\text{J}$

而误差计算公式为：

$(357939.5-347021.5)/357939.5\approx3.0502\%$

其余作业及阶段性任务能耗实测值能耗预测值及误差如表5所列。

表5 能耗模型预测值及误差

| 作业名称 | 阶段性任务 | 能耗实测值（单位：J） | 能耗模型预测值（单位：J） | 误差 | $\sigma=1.09$时的能耗预测值 | $\sigma=1.09$时的误差 |
|---|---|---|---|---|---|---|
| WordCount | N/A | 357939.5 | 347021.5 | 3.0502% | 378253.435 | −5.6752% |
| TeraSort | N/A | 245815.25 | 221722 | 9.8014% | 241676.98 | 1.6835% |
| NutchIndex | N/A | 302075 | 285485.6 | 5.4918% | 311179.304 | −3.0139% |
| K-means | Cluster Iterator | 54958 | 52426.35 | 4.6065% | 57144.7215 | −3.9789% |
| | Cluster Classification | 105560 | 97034.7 | 8.0763% | 105767.823 | −0.1969% |
| PageRank | Stage1： | 143036.25 | 136556.75 | 4.5300% | 148846.8575 | −4.0623% |
| | Stage2： | 69097 | 61764 | 10.6126% | 67322.76 | 2.5678% |
| Bayes | DocumentProcessor:DocumentTokeuizer | 5082 | 4452 | 12.3967% | 4852.68 | 4.5124% |
| | CollocDriver.generateCollocations | 165659.25 | 158026.5 | 4.6075% | 172248.885 | −3.9778% |
| | CollocDriver.computeNGrams | 5968.575 | 5846.3 | 2.0486% | 6372.467 | −6.7670% |
| | DictionaryVectorizer:MakeParitialVectors | 6469.5 | 6155.75 | 4.8497% | 6709.7675 | −3.7138% |
| | ParticalVectorMerger:MergePartialVectors | 1071.5 | 892.85 | 16.6729% | 973.2065 | 9.1734% |
| | VectorTfld Document Frequency Count | 1047.5 | 871.9 | 16.7637% | 950.371 | 9.2725% |
| | MakePratialVectors | 1083.5 | 919.65 | 15.1223% | 1002.4185 | 7.4833% |
| | ParticalVectorMerger:MergePartialVectors | 1012 | 910.95 | 9.9852% | 992.9355 | 1.8838% |
| | TrainNaiveBayesJob IndexInstancesMapper-Reducer | 1118 | 1000.25 | 10.5322% | 1090.2725 | 2.4801% |
| | TrainNaiveBayesJob WeightsMapper-Reducer | 993 | 859.1 | 13.4844% | 936.419 | 5.6980% |

从表5中的数据可以发现，能耗模型预测值总是比实际测量值小。造成这样结果的原因主要有两个：1）预测能耗模型在任务分解及调度结果确定后，进行能耗估算时没有考虑到任务的执行失败问题，而在任务的实际运行过程中存在任务的失败现象，在对失败任务重调度及重新执行过程中消耗了额外的能耗，使得估算值小于实际测量值；2）任务之间存在等待现象时将耗费掉额外的能耗，造成实际能耗值大于预测能耗值。观察表5可以发现很多本身存在误差较大（大于10%）的作业，这是因为任务执行时间越短，任务数量越多，估算误差越大，较短的任务执行时间使得能耗采样数据较小，导致能耗测量值本身的误差。

因此基于以上事实，本文采用能耗预测准确率调节因子来对能耗预测模型的准确率进行修正。当将所有作业的因子值都设置为1.09时（实际运用过程中，不同作业的因子可不同），可将所有作业的能耗预测误差率控制在10%以内，如表5所列。

**结束语** 在能源价格上涨、数据中心规模不断扩大的今天，高能耗已逐渐成为制约大数据快速发展的一个主要瓶颈，研究MapReduce能耗模型具有重要意义。已有研究主要通过选择部分节点执行任务、任务完成后关闭节点、配置参数优化、DVFS调度、作业调度、虚拟机放置策略及数据压缩等方法达到提高MapReduce能耗利用率的目的。而本文主要针对MapReduce能耗模型，在对MapReduce的集群结构、作业的任务分解及任务与资源映射模型分析建模的基础上，提出基于作业历史运行信息的MapReduce能耗预测模型，实现在MapReduce作业执行前对作业能耗进行预测，为节能的MapReduce任务调度研究打下基础。大量的实验及能耗数据分析表明，本文提出的能耗预测模型具有有效性，并且通过对能耗预测准确率调节因子的不断修正，能够提高能耗模型的预测准确度，使得能耗预测模型更加可行。

下一步工作主要有两个方面：1）基于本文提出的能耗预测模型，开发适配Hadoop的能耗预测插件，并研究根据作业运行历史记录自动提取模型参数值的方法；2）能耗预测模型提供了作业调度后作业运行前对能耗进行计算的能力，为研究节能的MapReduce调度算法提供了基础。

## 参 考 文 献


[1] 孟小峰，慈祥. 大数据管理：概念、技术与挑战[J]. 计算机研究与发展，2013，50(1)：146-149
Meng X F, Ci X. Big Data Management: Concepts, Techniques and Challenges[J]. Journal of Computer Research and Development, 2013, 50(1): 146-149

[2] Gantz J, Chute C, Manfrediz A, et al. The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011 [EB/OL]. 2013-5-25. http://wwww.if-ap.ru/ library/book268.pdf

[3] Global action plan, an inefficient truth [EB/OL]. 2007 . 2011-02-12. http://globalactionplan. org. uk

[4] Times N Y. Power, Pollution and the Internet [EB/OL]. 2013-5-20. http://www. nytimes. com/2012/09/23/technology/ data-ceneters-waste-vast-amounts-of-energy-belying-industry-image. html

[5] Dean J, Ghemawat S. MapReduce: Simplifed data processing on large clusters[C]// Proceedings of the Conference on Operating System Design and Implementation (OSDI). New York: ACM, 2004:137-150

[6] Barroso L A, Hlzle U. The datacenter as a computer: An introduction to the design of warehouse-scale machines [R]. Morgan: Synthesis Lectures on Computer Architecture, Morgan & Claypool Publishers, 2009

[7] 王鹏,孟丹,詹剑锋,等. 数据密集型计算编程模型研究进展[J]. 计算机研究与发展,2010,47(11):1993-2002
Wang P, Meng D, Zhan J F, et al. Review of Programming models for data-Intensive computing[J]. Journal of Computer Research and Development, 2010, 47(11):1993-2002

[8] Li D, Wang J E. Energy efficient redundant and inexpensive disk array [C]// Proceedings of the ACM SIGOPS European Workshop. New York: ACM, 2004:29-35

[9] Albers S. Energy-efficient algorithms [J]. Communications of the ACM, 2010, 53(5):86-96

[10] Wierman A, Andrew L L, Tang A. Power-aware speed scaling in processor sharing systems [C]// Proceedings of the 28th Conference on Computer Communications (INFOCOM 2009). Piscataway, NJ, IEEE, 2009:2007-2015

[11] Andrew L L, Lin M, Wierman A. Optimality, fairness, and robustness in speed scaling designs [C]// Proceedings of ACM International Conference on Measurement and Modeling of International Computer Systems (SIGMETRICS 2010). New York: ACM, 2010:37-48

[12] Meisner D, Gold B T, Wenisch T F. PowerNap: Eliminating server idle power [J]. ACM SIGPLAN Notices, 2009, 44(3): 205-216

[13] Choi J, Govindan S, Jeong J, et al. Power consumption prediction and power-aware packing in consolidated environments [J]. IEEE Transactions on Computers, 2010, 59(12):1640-1654

[14] Liao X, Jin H, Liu H. Towards a green cluster through dynamic remapping of virtual machines[J]. Future Generation Computer Systems, 2012, 28(2):469-477

[15] Jang J W, Jeon M, Kim H S, et al. Energy reduction in consolidated servers through memory-aware virtual machine scheduling [J]. IEEE Transactions on Computers, 2011, 99(1):552-564

[16] Wang X, Wang Y. Coordinating power Control and performance management for virtualized server cluster[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(2):245-259

[17] Wang Y, Wnag X, Chen M, et al. Partic: Power-aware response time control for virtualized web servers[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(2):323-336

[18] Garg S K, Yeo C S, Anandasivam A, et al. Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers [J]. Journal of Parallel and Distributed Computing, 2010, 71(6):732-749

[19] Kusic D, Kephart J O, Hanson J E, et al. Power and performance management of virtualized computing environments via lookahead control [J]. Cluster Computing, 2009, 12(1):1-15

[20] Gmach D, Rolia J, Cherkasova L, et al. Resource pool management: Reactive versus proactive or let's be friends[J]. Computer Networks, 2009, 53(17):2905-2922

[21] 廖彬,于炯,张陶,等. 基于分布式文件系统 HDFS 的节能算法[J]. 计算机学报,2013,36(5):1047-1064
Liao B, Yu J, Zhang T, et al. Energy-Efficient Algorithms for Distributed File System HDFS[J]. Chinese Journal of Computers, 2013, 36(5):1047-1064

[22] 廖彬,于炯,孙华,等. 基于存储结构重配置的分布式存储系统节能算法[J]. 计算机研究与发展,2013,50(1):3-18
Liao B, Yu J, Sun H, et al. Energy-Efficient Algorithms for Distributed Storage System Based on Data Storage Structure Reconfiguration[J]. Journal of Computer Research and Development, 2013, 50(1):3-18

[23] 廖彬,于炯,钱育蓉,等. 基于可用性度量的分布式文件系统节点失效恢复算法[J]. 计算机科学,2013,40(1):144-149
Liao B, Yu J, Qian Y R, et al. The Node Failure Recovery Algorithm for Distributed File System based on Measurement of Data Availability[J]. Computer Sicence, 2013, 40(1):144-149

[24] 廖彬,于炯,张陶,等. 一种适应节能的云存储系统元数据动态建模与管理方法[J]. 小型微型计算机系统,2013,10(34):2407-2412
Liao B, Yu J, Zhang T, et al. A Novel Energy-efficient Metadata Dynamic Modeling and Management Approach for Cloud Storage System[J]. Journal of Chinese Computer Systems, 2013, 10(34):2407-2412

[25] Leverich J, Kozyrakis C. On the energy(in)efficiency of hadoop clusters [J]. ACM SIGOPS Operating Systems Review, 2010, 44(1):61-65

[26] Lang W, Patel J M. Energy management for mapreduce clusters [J]. Proceedings of the VLDB Endowment, 2010, 3(1/2):129-139

[27] Chen Y, Keys L, Katz R H. Towards energy effcient mapreduce [R]. Berkeley: EECS Department, University of California, 2009

[28] Wirtz T, Ge R. Improving MapReduce energy efficiency for computation intensive workloads [C] // 2011 International Green Computing Conference and Workshops (IGCC). IEEE, 2011:1-8

[29] Goiri i, Le K, Nguyen T D, et al. GreenHadoop: leveraging green energy in data-processing frameworks[C]// Proceedings of the 7th ACM European Conference on Computer Systems. ACM, 2012:57-70

[30] Cardosa M, Singh A, Pucha H, et al. Exploiting Spatio-Temporal Tradeoffs for Energy Efficient MapReduce in the Cloud[D]. Department of Computer Science and Engineering, University of Minnesota, 2010

[31] Chen Y, Ganapathi A, Katz R H. To Compress or Not to Compress-Compute vs. IO Tradeoffs for Mapreduce Energy Efficiency[C]// Proceedings of the First ACM SIGCOMM Workshop on Green Networking. New Delhi, India, 2010:23-28

[32] 宋杰,李甜甜,朱志良,等. 云数据管理系统能耗基准测试与分析[J]. 计算机学报,2013,36(7):1485-1499
Song J, Li T T, Zhu Z L, et al. Benchmarking and analyzing the energy consumption of cloud data management system [J]. Chinese Journal of Computers, 2013, 36(7):1485-1499