



文章编号:1001-9081(2019)01-0205-08

DOI:10.11772/j.issn.1001-9081.2018051055

## 基于 SSD 数据库负载的 SQL 能耗感知模型

李树<sup>1</sup>, 于炯<sup>1,2\*</sup>, 国冰磊<sup>2</sup>, 蒲勇霖<sup>2</sup>, 杨德先<sup>2</sup>, 刘粟<sup>2</sup>

(1. 新疆大学 软件学院, 乌鲁木齐 830008; 2. 新疆大学 信息科学与工程学院, 乌鲁木齐 830046)

(\* 通信作者电子邮箱 yujiong@xju.edu.cn)

**摘要:**面对大数据带来的能耗及环境方面的严峻问题,构建节能的绿色数据库系统已成为关键需求和重要挑战。针对现有数据库系统主要以性能优化为目标,缺少对能耗的感知及优化的问题,提出基于数据库负载的能耗感知模型,并将模型应用于基于固态硬盘(SSD)的数据库系统中。首先,将数据库负载执行过程中对主要系统资源(CPU、固态硬盘)的消耗解析为时间开销和功耗开销,并基于 SSD 数据库负载的基本 I/O 类型构建时间开销模型和功耗开销模型,实现为数据库构建资源开销单位统一的能耗感知模型;然后,利用多元线性回归实现对模型的求解,并分别在独占环境和竞争环境下,验证模型对不同 I/O 类型的数据库负载能耗估算的准确性;最后,分析实验结果,并讨论了影响模型准确性的因素。经实验验证模型准确度较高,在 DBMS 独占系统资源情况下的平均误差为 5.15%,绝对误差不超过 9.8%;竞争环境下的准确率相对下降,但平均误差也低于 12.21%,可有效构建能耗感知的绿色数据库系统。

**关键词:**能耗感知模型; 固态硬盘; 绿色计算

**中图分类号:** TP315    **文献标志码:**A

### SQL energy consumption perception model for database load based on SSD

LI Shu<sup>1</sup>, YU Jiong<sup>1,2\*</sup>, GUO Binglei<sup>2</sup>, PU Yonglin<sup>2</sup>, YANG Dexian<sup>2</sup>, LIU Su<sup>2</sup>

(1. School of Software, Xinjiang University, Urumqi Xinjiang 830008, China;

2. School of Information Science and Engineering, Xinjiang University, Urumqi Xinjiang 830046, China)

**Abstract:** For energy consumption and severe environmental problems brought by big data, building an energy-efficient green database system has become a key requirement and an important challenge. To solve the problem that traditional database systems mainly focus on performance, and are lack of energy consumption perception and optimization, an energy consumption perception model based on database workload was proposed and applied to the database system based on Solid-State Drive (SSD). Firstly, the consumption of major system resources (CPU, SSD) during database workload execution was quantified as time overhead and power consumption overhead. Based on basic I/O type of SSD database workload, a time cost model and a power consumption overhead model were built, and an energy consumption perception model with uniform resource unit was implemented. Then, multi-variable linear regression mathematical tools were used to solve the model, and in the exclusive environment and competitive environment, the energy estimation accuracy of the model for different I/O types of database workload was verified. Finally, the experimental results were analyzed and the factors that affect the model accuracy were discussed. The experimental results show that the model accuracy is relatively high. Under ideal conditions that DBMS monopolized system resources, the average error is 5.15% and the absolute error is no more than 9.8%. Although the accuracy in competitive environment is reduced, the average error is less than 12.21%. The model can effectively build an energy-aware green database system.

**Key words:** energy consumption perception model; Solid-State Drive (SSD); green computing

## 0 引言

数据中心作为信息存储的重要载体,其数量和建设规模呈爆炸性增长<sup>[1]</sup>,而由此产生的巨量能源消耗所带来的能耗及环境问题日益严峻。据报道,美国 2011 年的电网总量中,仅数据中心产生的能耗就达到了 2%<sup>[2]</sup>,同年,我国数据中心

的总耗电量为 700 亿千瓦时(kWh),占总耗电量的 5%<sup>[3]</sup>。此外,美国纽约时报<sup>[4]</sup>也指出全球数据中心一年的用电总量高达 3000 亿瓦特,几近于 30 座核发电站的产电总量,其中却仅有 6%~12% 的能耗真正用于处理用户请求。与此同时,巨量的能耗带来的碳排放问题也相继引发了一系列环境和社会问题。就信息技术领域而言,其产生的二氧化碳排放量就

收稿日期:2018-05-22;修回日期:2018-07-17;录用日期:2018-07-23。

基金项目:国家自然科学基金资助项目(61462079, 61562078, 61562086); 国家科技支撑项目(2015BAH02F01)。

**作者简介:**李树(1993—),男,安徽蚌埠人,硕士研究生,CCF 会员,主要研究方向:绿色计算、机器学习; 于炯(1964—),男,北京人,教授,博士生导师,博士,CCF 高级会员,主要研究方向:网络安全、网格计算、分布式计算; 国冰磊(1991—),女,湖北襄阳人,博士研究生,CCF 会员,主要研究方向:绿色计算; 蒲勇霖(1991—),男,山东淄博人,博士研究生,CCF 会员,主要研究方向:绿色计算、分布式计算; 杨德先(1991—),男,新疆塔城人,博士研究生,CCF 会员,主要研究方向:绿色计算; 刘粟(1994—),女,吉林吉林人,硕士研究生,CCF 会员,主要研究方向:分布式计算、内存计算。



约占全球总量的 2%，且这一比例到 2020 年仍将翻一番<sup>[5]</sup>。由于数据库负载占用着数据中心大量服务器资源，使得 DBMS 成为数据中心总能耗的重要组成部分<sup>[6]</sup>，因此，针对节能的绿色数据库系统的研究是解决当前能耗及环境问题的关键，具有显著的应用价值和社会意义。

目前，由于固态硬盘 (Solid-State Drive, SSD) 具有高性能、低能耗等特点，因此为构建低能耗的绿色数据库系统提供了新思路，许多相关研究学者也相继提出了在绿色数据库系统中应当考虑 SSD 的使用<sup>[7-8]</sup>，但现有针对基于 SSD 的数据库系统的研究主要以提高性能为目标，缺少对系统能耗的感知与处理<sup>[9-10]</sup>。与此同时，现有数据库系统的查询计划往往是基于机械磁盘 (Hard Disk Drive, HDD) 进行设计与优化，简单用 SSD 替换 HDD，并不能够选择最有效地权衡性能和能耗的执行计划，因此，本文基于 SSD 数据库系统，以降低系统能耗为首要目标，重点关注如何有效估算执行计划的能耗，进而优化 DBMS 的 SQL 语句执行计划的选择来实现数据库节能。通过对固态硬盘顺序读、随机读、顺序写和随机写的不同操作进行区分，建立基于 SSD 数据库不同 I/O 操作类型的 SQL 语句能耗感知模型。模型只需要从 SQL 语句的执行计划中获取 CPU 指令总数、固态硬盘读写次数等资源消耗信息，通过相应算法就可得出 CPU、固态硬盘等主要能耗部件的能量消耗，最终估算出相应执行计划在系统中的总能耗。本文主要工作是在对直接影响数据库能耗的 SQL 资源消耗分析基础上，为基于 SSD 的数据库系统构建了基于数据库 I/O 操作类型、单位开销统一的能耗感知模型，通过实验验证模型的有效性，并进行了优化分析。实验结果表明，提出的能耗感知模型在理想情况下，平均误差为 5.15%，绝对误差不超过 9.8%，证明了该模型可有效应用于构建能耗感知的绿色节能数据库系统。模型主要基于读写两种不同操作类型，将 CPU 和 SSD 的资源开销统一转化为顺序读或顺序写的单位资源开销，在查询优化器中仅通过读写次数就可预测资源开销和能量消耗。

## 1 相关研究

近年来，构建节能的数据库系统已成为学术界和工业界的共识，现有工作主要集中在硬件和软件两个方面。硬件方面侧重于高性能、低功耗硬件（如处理器、存储器）的设计和使用，以构建低能耗的绿色硬件系统为目标。目前利用固态硬盘替换磁盘是该方向的研究热点。吕雁飞等<sup>[11]</sup>测试了 SSD 基本 IO 特性，并分析了 CPU 处理能力、缓冲区大小等对基于 SSD 的数据库的影响，最后从数据组织以及数据库资源利用等多方面对基于 SSD 的数据库系统进一步提高性能、降低能耗给出优化建议。Bausch 等<sup>[12]</sup>基于 PostgreSQL 数据库对查询处理性能进行了对比测评，相比 HDD，SSD 环境下查询处理性能最高提升了近 50 倍，有效降低了系统能耗，但不同扫描操作的性能提升幅度差距较大。Do 等<sup>[13]</sup>利用 SSD 替换 HDD 对循环嵌套、排序—合并以及 Hash 等连接算法的性能进行了测试。实验结果显示连接操作在 SSD 上的执行时间均低于 HDD，系统能耗也得到了降低，但性能提升幅度不足 2 倍。Lee 等<sup>[14]</sup>测试了基于 SSD 数据库的多方面性能和功耗

表现，并探讨了使用 SSD 后的数据库系统的性能瓶颈是否会由存储设备转移到 CPU 的问题。以上基于硬件的节能研究主要通过降低执行时间的方式实现了一定的节能效果。

软件方面主要研究基于功耗感知的查询优化器。首先针对系统构建功耗/能耗模型，然后结合模型和查询优化器，在满足用户服务等级协议 (Service Level Agreement, SLA) 的基础上，最终为查询语句选择具有较低功耗/能耗的执行计划。Rodriguez-Martinez 等<sup>[15]</sup>通过统计不同执行计划的元组大小、基数和列数信息，并利用系统负载数据和内部传感器数据，建立查询语句级别的功耗模型。模型准确性较高，但仅局限于查询语句，不能完全覆盖所有数据库操作类型。Xu 等<sup>[16-18]</sup>根据数据元组数、索引元组数和读取页面数等运算符特征，建立运算符级别的功耗模型，并结合性能约束和功耗开销双重标准来评估查询计划的优劣，从而选择性能良好、相对节能的查询计划，但模型准确度和稳定性较差。杨良怀等<sup>[19]</sup>和陈俊等<sup>[20]</sup>通过收集 CPU 所有核心的利用率、执行频率、磁盘利用率以及相应系统功耗数据，借助多元线性回归对收集的信息进行拟合，得到功耗预测模型。模型全面地兼顾了 CPU 各个核对系统总功耗的影响，具有较高的应用价值。国冰磊等<sup>[21-22]</sup>基于磁盘的传统关系型数据库，通过对数据库 SQL 语句主要硬件 (CPU、HDD) 的资源消耗信息进行线性回归，构建了单位代价统一的动态功耗预测模型，模型的准确度较高且稳定性良好。

然而，现有针对基于 SSD 的数据库的研究与设计，往往以提高性能为目标，较少关注能耗优化；同时，现有的节能查询优化器主要是基于 HDD 进行设计与优化，SSD 相比 HDD 不仅性能优异，也有读写不对称、写前擦除等缺点，因此将针对基于 HDD 的数据库设计的能耗优化算法直接迁移到基于 SSD 的数据库中，并不能有效利用 SSD 的特性。本文根据 SSD 随机读写和顺序读写的性能和能耗不同的特点，通过对 SQL 语句执行计划的 CPU 指令数及 SSD 读写次数等进行分析，建立可以估算 SQL 执行计划资源消耗的能耗感知模型。将模型植入到基于 SSD 的数据库查询优化器中，选择能够较好权衡性能和能耗的执行计划，从而为构建能耗感知的绿色数据库系统奠定基础。本文与已有工作的不同之处在于：

- 1) 现有针对基于 SSD 的数据库系统的研究主要致力于系统性能的提升，而本文主要目标是在降低数据库系统能耗的同时兼顾性能；
- 2) 能耗感知模型对 CPU 部件能耗的估算采用能更好反映 CPU 工作量的执行指令总数，相比相关研究选取 CPU 处理元组数度量能耗的方式，本文模型的准确度更高。
- 3) 现有模型主要针对查询语句（主要是读操作）进行能耗优化。本文基于数据库 I/O 操作进行建模，兼顾读写两种操作类型，可以对不同类型的 SQL 语句进行较全面的覆盖。

## 2 能耗感知模型

SQL 语句是连接并操作现有关系型数据库系统的标准接口，同时数据库资源的 70% ~ 90% 都消耗在了 SQL 语句的执行过程中。SQL 语句的执行时间和资源消耗直接影响着数据库负载的性能和能耗，因此本文对 SQL 语句的能耗进行建模



与优化可以达到预测数据库系统能耗的目的,对提高系统资源利用率和实现构建节能的绿色数据库系统具有重大研究意义。

能耗度量是能耗感知的基础,根据经典能耗公式可知,系统能耗即系统功率与执行时间的乘积。设系统能耗为  $E$ (单位:J)、系统功率为  $P$ (单位:W)(采用平均功率计算)、执行时间为  $t$ (单位:s),则能耗计算公式可表示为:

$$E = \int_t^{t+\Delta t} P dt \quad (1)$$

设任意一条SQL语句成功执行后的能耗为  $E_{SQL}$ 。在基于SSD的数据库系统中,每条SQL语句在执行过程中都主要消耗CPU、存储设备SSD、内存以及网络等资源,因此一条SQL语句执行完毕后的能耗可由式(2)得到:

$$E_{SQL} = E_{CPU} + E_{MEM} + E_{SSD} + E_{NET} + E_{Other} \quad (2)$$

其中: $E_{CPU}$ 、 $E_{MEM}$ 、 $E_{SSD}$ 、 $E_{NET}$ 分别为CPU、内存、固态硬盘及网络等资源部件的能耗, $E_{Other}$ 为消耗其他资源产生的能耗。对SQL语句而言,CPU和存储设备是主要的能耗部件,因此本文能耗感知模型主要考虑CPU和SSD的能耗。考虑到优化其他能耗部件对于降低系统总能耗的效果不显著,故本文将内存、网络资源等能耗部件视为系统静态能耗处理。

结合能耗公式可知,在基于SSD的数据库负载执行过程中,对于CPU和存储设备SSD而言,都会产生时间开销和功耗开销。本文分别针对时间开销和功耗开销进行建模,最终得到可以估算DBMS资源消耗的能耗感知模型。模型中都包含CPU开销和SSD上的I/O开销,通过对以下资源开销量化,得到单位统一的资源开销模型。

1)CPU开销。在SQL语句执行过程中,CPU的工作主要包括处理元组、解析语句、建立查询树、执行计划等,仅利用处理的元组数评价CPU开销较为粗糙<sup>[17]</sup>,因此选择更能实际反映CPU工作量的指令总数对CPU时间开销和功耗开销进行评估。

2)固态硬盘开销。固态硬盘产生的时间开销和功耗开销都与其产生的读写数据量有关。在基于SSD的数据库系统中,SQL语句执行过程中常见的I/O类型主要包括:顺序读、顺序写、随机读、随机写。例如,对Oracle而言,其访问存储设备的最小单位是数据块,其SQL语句的I/O操作主要为顺序读数据块、随机读数据块、顺序写数据块、随机写数据块。采用读写数据块的数量作为固态硬盘资源消耗的计量单位,可以较好地反映固态硬盘的工作量。

## 2.1 时间开销预测模型

SQL语句的时间开销主要由CPU时间开销和固态硬盘的I/O时间开销构成。在数据库系统中,一个常用统计参数Cost(%CPU),为在总时间开销中,CPU时间开销所占的比重。由于实验中的SQL语句结构简单,执行时间较短,所以CPU时间开销相比I/O时间开销而言可以忽略不计,则总的时间开销就等于I/O时间开销:

$$T = T_{IO} \quad (3)$$

根据I/O不同读写类型,将不同类型读写操作的次数乘以该操作的平均单次处理时间定义为总的时间开销。对于I/O时间开销而言,在不同数据库系统配置下,数据块读写操

作有不同的操作时间。为统一时间开销,利用顺序读、顺序写的平均单次时间开销分别作为读、写操作的最长时间开销单位,I/O类型为顺序读、顺序写的总时间开销即分别为顺序读、顺序写的总次数。

设  $s_r_t$  为单次顺序读的平均时间(单位:ms), $s_r_sm$  为顺序读的总次数, $r_r_t$  为单次随机读的平均时间(单位:ms), $r_r_sm$  为随机读总次数; $s_w_t$  为单次顺序写的平均时间(单位:ms), $s_w_sm$  为顺序写的总次数; $r_w_t$  为单次随机写的平均时间(单位:ms), $r_w_sm$  为随机写的总次数。

设顺序读操作的时间开销为  $T_{s_r}$ ,公式为:

$$T_{s_r} = s_r_sm \quad (4)$$

随机读操作的时间开销为  $T_{r_r}$ ,公式为:

$$T_{r_r} = (r_r_t / s_r_t) * r_r_sm \quad (5)$$

顺序写操作的时间开销为  $T_{s_w}$ ,公式为:

$$T_{s_w} = s_w_sm \quad (6)$$

随机写操作的时间开销为  $T_{r_w}$ ,公式为:

$$T_{r_w} = (r_w_t / s_w_t) * r_w_sm \quad (7)$$

其中,式(5)、(7)将随机读写的时间开销转换为顺序读写的时间开销;同时,各类型I/O操作实际的时间开销可以利用式(4)或(5)与  $s_r_t$  相乘,式(6)或(7)与  $s_w_t$  相乘获得。

## 2.2 功耗开销预测模型

系统总功耗主要包括系统在空闲状态下的静态功耗和DBMS运行负载情况下CPU和I/O操作产生的动态功耗。由于静态功耗为固定值,因此功耗开销预测模型主要关注估算负载情况下的CPU和I/O产生的动态功耗之和。同样,利用顺序读写的功耗开销统一CPU和I/O功耗开销:

$$P_{SQL} = P_{CPU} + P_{IO} \quad (8)$$

### 2.2.1 I/O功耗开销

I/O操作类型主要为顺序读、随机读、顺序写和随机写,同样利用一次顺序读和顺序写所花费的功耗开销将所有功耗开销统一,因此数据库读的功耗开销都会转换为顺序读的功耗开销,数据库写的功耗开销都会转换为顺序写的功耗开销。

设  $s_r_p$  为单次顺序读的平均功率(单位:W), $s_r_sm$  为顺序读的总次数; $r_r_p$  为单次随机读的平均功率(单位:W), $r_r_sm$  为随机读总次数; $s_w_p$  为单次顺序写的平均功率(单位:W), $s_w_sm$  为顺序写的总次数; $r_w_p$  为单次随机写的平均功率(单位:W), $r_w_sm$  为随机写的总次数。

设顺序读操作的功耗开销为  $P_{s_r}$ ,公式为:

$$P_{s_r} = s_r_sm \quad (9)$$

随机读操作的功耗开销为  $P_{r_r}$ ,公式为:

$$P_{r_r} = (r_r_p / s_r_p) * r_r_sm \quad (10)$$

顺序写操作的功耗开销为  $P_{s_w}$ ,公式为:

$$P_{s_w} = s_w_sm \quad (11)$$

随机写操作的功耗开销为  $P_{r_w}$ ,公式为:

$$P_{r_w} = (r_w_p / s_w_p) * r_w_sm \quad (12)$$

各类型I/O操作的实际功耗开销可以利用式(9)或(10)与  $s_r_p$  相乘,式(11)或(12)与  $s_w_p$  相乘获得。

### 2.2.2 CPU功耗开销

在SQL语句执行过程中,CPU会根据不同的数据处理量和操作方式,执行不同数量的CPU指令,消耗的所有CPU指



令称为总指令数( $CPU\_num$ )。为计算方便,CPU 总指令数以万为计量单位。为了统一 CPU 功耗开销和 IO 功耗开销,将  $CPU\_to\_IO$  定义为转换参数,其中  $CPU\_to\_IO$  为一次顺序读的能耗  $s\_r\_p$  与等价的 CPU 指令数能耗  $P\_CPU\_unit$  之间的转换。

$$P\_CPU = CPU\_num/CPU\_to\_IO \quad (13)$$

转换参数的计算如下:

$$CPU\_to\_IO = P\_CPU\_unit * s\_r\_p \quad (14)$$

由于不同型号 CPU 的计算能力差异较大,即单位时间能够计算完成的指令数不同,因此需要通过实验进行建模训练,从而获得  $P\_CPU\_unit$ ,即 CPU 的指令功耗能力。

则统一开销单位的 CPU 功耗开销为:

$$\begin{aligned} P\_CPU &= CPU\_num/CPU\_to\_IO = \\ &CPU\_num/(P\_CPU\_unit * s\_r\_p) \end{aligned} \quad (15)$$

因此,I/O 类型是顺序读的功耗开销为:

$$\begin{aligned} P\_SQL &= P\_IO + P\_CPU = \\ &s\_r\_sm + CPU\_num/(P\_CPU\_unit * s\_r\_p) \end{aligned} \quad (16)$$

I/O 类型是随机读的总功耗开销为:

$$\begin{aligned} P\_SQL &= P\_IO + P\_CPU = (r\_r\_p/s\_r\_p) * s\_r\_sm + \\ &CPU\_num/(P\_CPU\_unit * s\_r\_p) \end{aligned} \quad (17)$$

I/O 类型是顺序写的总功耗开销为:

$$\begin{aligned} P\_SQL &= P\_IO + P\_CPU = s\_w\_sm + \\ &CPU\_num/(P\_CPU\_unit * s\_r\_p) \end{aligned} \quad (18)$$

I/O 类型是随机写的总功耗开销为:

$$\begin{aligned} P\_SQL &= P\_IO + P\_CPU = (r\_w\_p/s\_w\_p) * s\_w\_sm + \\ &CPU\_num/(P\_CPU\_unit * s\_r\_p) \end{aligned} \quad (19)$$

设数据库系统总功耗开销为  $P$ ,数据库系统的总能耗为  $E$ ,系统在空闲状态下的功耗为  $P_{free}$ ,则系统的总功耗开销和总能耗开销如下:

$$P = P\_SQL + P_{free} \quad (20)$$

$$E = (P\_SQL + P_{free}) * T\_IO \quad (21)$$

结合以上时间开销和功耗开销模型即得到数据库系统的能耗感知模型,但实验需要训练出 CPU 指令功耗能力以及数据库各类型 I/O 操作的平均单次执行功耗。以数据库顺序读的模型求导为例,综合式(16)、(20)得到 I/O 类型为数据库顺序读的系统总功耗如式(22)所示:

$$\begin{aligned} P &= P_{free} + s\_r\_p * s\_r\_sm + \\ &(1/P\_CPU\_unit) * CPU\_num \end{aligned} \quad (22)$$

为方便求解,将顺序读的总能耗公式转换为数学公式,设系统总能耗  $P$  为  $y$ , $CPU\_num$  为  $X_1$ , $s\_r\_sm$  为  $X_2$ ,得到以下线性回归方程:

$$y = \alpha_0 + \alpha_1 * X_1 + \alpha_2 * X_2 \quad (23)$$

其中: $X_1$ 、 $X_2$  为回归变量, $\alpha_0$ 、 $\alpha_1$ 、 $\alpha_2$  为回归系数。这里, $\alpha_0$  代表系统静态功耗, $\alpha_1$  代表每万条 CPU 指令的功耗开销, $\alpha_2$  代表单次数据库顺序读的功耗开销。

### 3 实验设计与模型求解

#### 3.1 实验环境和数据收集

实验采用 HOPI 数字功耗仪收集功耗等数据,数据采集

频率设置为每秒 1 次。为排除用电监测软件(HP8713)对采样数据准确度的干扰,实验采用双机通信的方式,具体实验平台如图 1 所示。总体实验环境描述如表 1 所列。

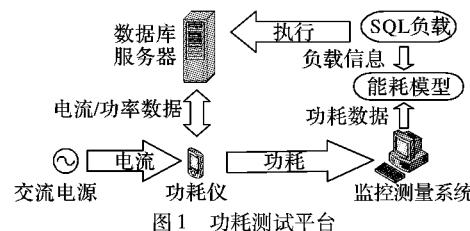


Fig. 1 Testing platform of power consumption

表 1 实验环境描述

Tab. 1 Description of experimental environment

项目	描述
操作系统	Windows 10 专业版 64 位
数据库系统	Oracle 11g
能耗数据测量	HOPI 待机功率测试仪 USB 智能版 (型号:HP-8713/2A 功率误差值 $\pm 0.01$ W)
能耗数据采集	用电监测仪数据分析系统 v1.0.5
能耗相关单位	功率:瓦特(W),能耗:焦耳(J),时间:秒(s)、毫秒(ms)
数据采样频率	每秒采集数据一次
主板	方正 Q87H3-AM
CPU	Intel Core i7-4790 CPU @ 3.60 GHz 八核
内存	8 GB 威刚 DDR3 1600 MHz
固态硬盘	Samsung SSD 850 EVO 250 GB

实验基于 Oracle 数据库平台,采用 TPC-H 测试基准,该基准由一系列面向商务应用的查询和并行数据修改组成。实验利用了 TPC-H 提供的 dbgen 工具生成 10 GB 大小的数据。

在 Oracle 数据库系统中,执行代价参数直接反映系统的处理能力,并提供了存储过程(GATHER\_SYSTEM\_STATS)收集这些相关统计数据。这些系统统计数据主要包括:CPU 执行指令总数( $CPU\_num$ )、数据库顺序读次数( $s\_r\_sm$ )、数据库随机读次数( $r\_r\_sm$ )、数据库顺序写次数( $s\_w\_sm$ )、数据库随机写次数( $r\_w\_sm$ )、单次数据库顺序读时间(Single block READ TIME, SREADTIME)、单次数据库随机读时间(Multi-block READ TIME, MREDTIME)等。这些统计数据为能耗感知模型提供了关键参数信息,对模型求解至关重要。其中,为尽可能提高能耗感知模型的精确度,参数 SREADTIME 和 MREDTIME 需要通过执行数据库负载进行大量训练后收集。

实验为减少误差,将每条 SQL 语句连续执行 100 遍来训练能耗感知模型,同时为了避免上一条 SQL 语句执行后产生的临时数据影响下一条 SQL 语句的执行,因此每次执行 SQL 语句之后都要清空缓冲区(BUFFER\_CACHE)和共享池(SHARED\_POOL)。Oracle 没有提供这种机制,因此利用存储过程(如表 2)实现以上功能,并最终在 PLSQL 中执行。

为保证 SQL 语句按照对能耗感知模型训练更有效的执行方式进行处理,运用了 Oracle Hints 技术强制 SQL 语句按照指定的方式去执行,但值得注意的是,Oracle Hints 在使用不当的情况下会失效。例如,当索引不为唯一非空索引时,则



关于索引的 Oracle Hints 操作会失效,结果将无法保证 SQL 按照指定的执行计划执行,因此实验必须保证 Oracle Hints 有效。

存储过程代码如下所示:

```
declare
i number:=0;
counts number;
begin
loop
i:=i + 1;
execute immediate 'alter system flush BUFFER_CACHE';
execute immediate 'alter system flush SHARED_POOL';
execute immediate '待测 SQL 语句';
bulk collect into t_table;
exit when i = 100;
end loop;
end loop;
```

### 3.2 数据库读操作能耗感知模型

在数据库中,常见执行数据库顺序读的操作主要有全表扫描和快速完全索引扫描两种,而常见执行数据库随机读的操作主要有索引完全扫描、索引范围扫描、索引跳跃扫描和由 ROWID 访问表四种,因此,采用这 6 类数据库读操作设计了两组数据集:第一组用于训练出能耗感知模型中的回归系数值,作为训练集;第二组用于验证能耗感知模型的实际效果和准确性,作为验证集。实验中训练集为 50 条,测试集为 10 条,它们的实例分别如表 2~3 所示。

表 2 读操作训练集实例

Tab. 2 Read operation training set instances

编号	操作类型	I/O 类型	SQL 例句
1	全表扫描	顺序读	select /* + full(orders) */ o_orderkey from orders;
2	索引完全扫描	随机读	select /* + index(orders sys_c0011162) */ o_orderkey from orders
3	索引跳跃扫描	随机读	select /* + index_ss(orders idx_orders) */ o_orderkey from orders

表 3 读操作测试集实例

Tab. 3 Read operation testing set instances

编号	操作类型	I/O 类型	SQL 例句
1	快速完全索引扫描	顺序读	select /* + index_ffs(orders sys_c0011162) */ o_orderkey from orders
2	索引范围扫描	随机读	select /* + index(orders sys_c0011162) */ o_orderkey from orders where o_orderkey >= 1 and o_orderkey <= 6000000
3	ROWID 访问表	随机读	select /* + rowid(orders) */ o_orderkey from orders where rowid >= 'AAASNZAABVJRAAA' and rowid <= 'AAASNZAABbEnAAJ'

实验使用功耗仪测得的功耗由数据库系统的静态功耗和动态功耗组成。静态功耗是系统在无工作负载状态下的功耗;动态功耗是数据库在执行 SQL 语句时产生的额外功耗,反映的是数据库系统资源消耗情况。在数据库处于无工作负载的空闲状态下,测得系统平均功耗为 29.8 W。在数据库独自占用整个系统资源的情况下运行训练集,并收集数据库系

统中的统计数据(模型所需参数及资源消耗等信息),利用 Matlab 软件将收集的数据代入建立的顺序读操作的功耗模型(23)中进行线性拟合,最终得到如下公式:

$$y = 34.74 + 0.0369X_1 + 0.3359X_2 \quad (24)$$

根据公式各项系数含义可知,实验对静态功耗的拟合结果为 34.74 W,与实际值(29.8 W)的相对误差为 16.5%。由于在基于 SSD 的数据库系统执行过程中,除 CPU 和固态硬盘会产生功耗以外,其他部件也会产生功耗;同时,系统还会产生一些无法预知的活动,因此其功耗高于静态功耗是合理的。实验收集到的单次数据库顺序读的时间(SREADTIM)为 1.26 ms,单次数据库随机读的时间(MREDTIM)为 2.73 ms。

结合拟合的功耗模型和 SREADTIM 值,可得顺序读操作的能耗模型为:

$$E = (34.74 + 0.0369X_1 + 0.3359X_2) * (X_2 * 1.26 \times 10^{-3}) \quad (25)$$

类似于数据库顺序读操作能耗模型的求导过程,利用收集到的统计数据与式(23)、式(27),同理可得数据库随机读操作的能耗感知模型为:

$$E = (34.74 + 0.0369X_1 + 0.945X_3) * (X_3 * 2.73 \times 10^{-3}) \quad (26)$$

其中: $X_1$  仍然表示 CPU 执行指令总数, $X_3$  表示随机读总次数( $r_{r\_sm}$ )。

由于数据库顺序读和随机读操作在数据库负载的实际执行过程中常常同时出现,所以为方便计算,使得模型更加通用,结合式(25)和(26)即得到数据库读操作的总能耗感知模型为:

$$E = \{ X_1 * (0.0465X_2 + 0.1007X_3) + 43.7724X_2 + 94.8402X_3 + 0.4232X_2^2 + 2.5799X_3^2 \} \quad (27)$$

### 3.3 数据库写操作的能耗感知模型

对于数据库写操作,常见且简单的操作类型主要为数据库插入(INSERT)和更新操作(UPDATE)。实验选取了执行顺序写操作和随机写操作的数据库插入语句和更新语句,同样设计了训练集(50 条)和测试集(10 条)两组数据集。数据库写操作的实例如表 4 所示。

表 4 写操作数据集实例

Tab. 4 Write operation dataset instances

编号	操作类型	I/O 类型	SQL 例句
1	插入	顺序写	insert into orders2 select * from orders
2	更新	顺序写	update orders_lineitem set o_shippriority = 1, l_linenumber = 1
3	更新	随机写	update orders2 t1 set t1.o_totalprice = (select t2.o_totalprice + 20 from orders2 t2 order by dbms_random.value)

在 DBMS 独自占用系统资源情况下,运行数据库写操作的训练集,收集系统统计数据,使用 Matlab 对数据库写操作的功耗模型进行线性拟合。由于实验设计的写操作往往要先执行数据库读操作以向内存加载写操作所需的数据,因此拟合过程中所需写操作的功耗数据必须为排除读操作影响的净功耗。实验首先根据写操作的执行时间  $T_w$  和平均功耗计算出



总能耗  $E_w$ ,然后利用收集的统计信息和读操作能耗感知模型的拟合公式(27)估算出读操作的能耗  $E_r$ ,最后通过如下计算公式获得写操作的净功耗  $P_w$ :

$$P_w = (E_w - E_r)/T_w \quad (28)$$

最终对数据库顺序写操作的拟合结果如下:

$$y = 35.46 + 0.0371X_4 + 0.4128X_5 \quad (29)$$

其中: $X_4$ 表示CPU指令总数, $X_5$ 表示顺序写操作次数。实验对静态功耗的拟合结果为35.46 W,与实际值(29.8 W)的相对误差为18.99%,高于数据库读操作对静态功耗的拟合结果。由于不仅其他部件会产生额外功耗,同时计算中还额外叠加了读操作能耗感知模型的误差。通过训练后,收集到的单次数据库顺序写的时间开销( $s_{w\_t}$ )为1.56 ms,单次数据库随机写的时间开销( $r_{w\_t}$ )为7.45 ms。

根据以上数据,利用线性拟合得到数据库顺序读操作的能耗感知模型的拟合公式为:

$$E = (35.46 + 0.0371X_4 + 0.4328X_5) * (X_5 * 1.56 \times 10^{-3}) \quad (30)$$

设数据库随机写次数( $r_{w\_sm}$ )为 $X_6$ ,将收集的参数代入随机写操作能耗感知模型中得到拟合公式为:

$$E = (35.46 + 0.0371X_4 + 1.5148X_6) * (X_6 * 7.45 \times 10^{-3}) \quad (31)$$

最终得出数据库写操作的通用能耗感知模型的公式为:

$$E = \{X_4 * (0.0579X_5 + 0.2764X_6) + 55.3176X_5 + 264.177X_6 + 0.6752X_5^2 + 11.2853X_6^2\} \quad (32)$$

## 4 模型验证及评估

实验对能耗感知模型的有效性和准确性进行了验证。能耗感知模型是基于数据库I/O类型构建的,因此实验所设计的测试集分别实现了顺序读、写和随机读、写共四种I/O操作,其测试集实例如表3所示。其中,每种操作的测试集包含10条语句,同时为验证模型的健壮性和适应性,分别在以下两种不同环境下对能耗感知模型进行验证评估。

1) 独占环境。系统仅运行DBMS,没有其他执行程序或软件与数据库系统竞争资源。在支持多任务的系统中,通常采用多处理级(Multi-Processing-Level, MPL)表示系统当前运行的软件及程序数,此时系统的多处理级为1。

2) 竞争环境。系统中存在和DBMS竞争资源的其他程序,实验中运行了Java程序,系统的MPL值为2。

以上四种类型的I/O操作主要可分为读、写两类;同时,模型在这两类操作上的结构设计和处理方式也具有明显区别,因此,为描述方便,下面的模型验证实验将围绕读、写两个方面展开。

### 4.1 读操作能耗感知模型的验证评估

为更客观地验证与评估读操作能耗感知模型对能耗的预测效果,本文还在相同实验环境下与文献[15~16]中提出的能耗模型(即参考模型)进行了对比实验。由图2(a)和(c)可知,在DBMS独占系统资源环境时,本文能耗感知模型的预测值相对于实际能耗的平均误差为5.15%,绝对误差不超过9.8%,参考模型的平均误差为6.55%,绝对误差不超过12.3%。由图2(b)、(d)可知,DBMS在与其他程序竞争系统

资源的条件下,由于系统无法预知的活动产生了更多资源消耗,使得模型在竞争环境下的误差高于独占环境。本文能耗感知模型在竞争环境下的平均误差为9.31%,绝对误差最高不超过18.9%,对应参考模型的平均误差为13.58%,绝对误差最高可达26.9%。总体而言,在DBMS独占系统资源环境时,本文建立的数据库读操作的能耗感知模型的预测准确度良好,相比文献[15~16]中提出的模型对数据库能耗的估算更加准确。由于本文能耗感知模型选用CPU指令数为计量单位,相比现有研究选用CPU元组数作为计量单位进行能耗估算的方式更加优越。

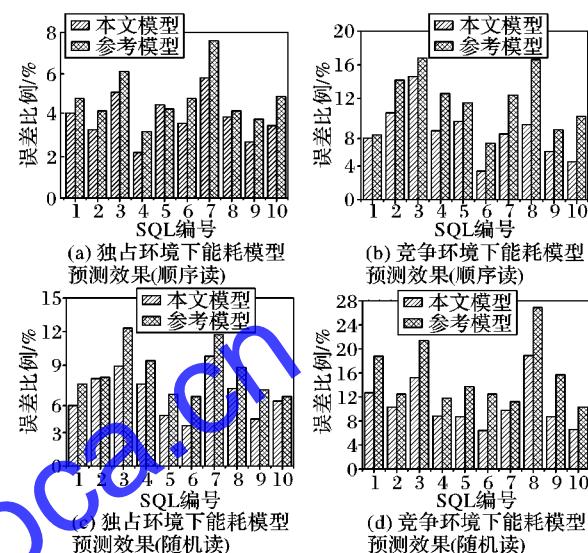


图2 读操作能耗感知模型验证效果

Fig. 2 Verification effects of reading operation energy perception model

### 4.2 写操作能耗感知模型的验证评估

图3所示为DBMS运行测试集时,本文写操作能耗感知模型的估算值相对于实际能耗的误差比例。由于现有相关研究主要针对查询语句即读操作进行能耗建模,没有关注写操作的能耗,因此本文在写操作能耗感知模型的验证实验中不涉及与其他模型的对照实验。不同于读操作,写操作的每组测试集都是由插入操作和更新操作两种操作组成。为使显示效果更加直观,每组写操作测试集的SQL编号1~5为插入操作,SQL编号6~10为更新操作。

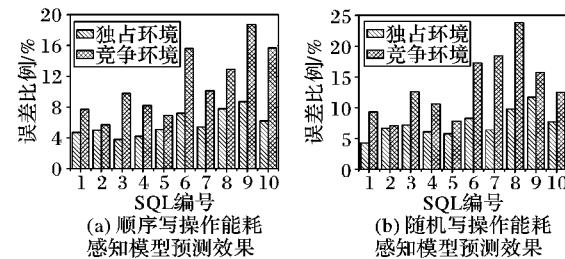


图3 本文写操作能耗感知模型验证效果

Fig. 3 Verification effects of proposed writing operation energy perception model

综合图3(a)、(b)可知,在DBMS处于独占环境时,能耗感知模型的估算值相对于实际能耗的平均误差为6.61%,绝对误差不超过11.7%;当DBMS处于竞争环境时,由于其他程序竞争系统资源,能耗感知模型对能耗的估算准确度明显



下降,平均误差为12.21%,绝对误差高达23.8%。值得注意的是,相比读操作,写操作能耗感知模型的误差较大,准确度明显降低。经分析,这一结果主要是由于更新操作的误差较大导致的。尤其在DBMS处于竞争环境执行插入操作时,能耗感知模型的平均误差为8.48%;而执行更新操作时,模型的平均误差高达15.94%。

#### 4.3 实验结果分析与优化

针对写操作的能耗感知模型,在估算更新操作能耗时的误差明显高于插入操作的实验结果,分析主要是由于SSD设备需要写前擦除的物理特性导致的。由于SSD不能原位更新,其在执行更新操作对数据进行重写时需要先执行擦除操作,而擦除操作是SSD资源开销代价最为昂贵的操作,其产生的额外功耗不可忽视。

同时,SSD这种写前擦除操作也是影响读操作能耗感知模型准确性和稳定性的重要因素。当读操作的数据量较大、内存资源相对紧张时,数据库会需要部分临时表空间来存储由于操作产生的临时数据,特别是数据库排序操作通常需要使用临时表空间,而临时表空间是数据库系统在存储设备上分配的一块独占区域,并向所有需要写临时表的操作开放共享,因此在向临时表空间写入数据时,往往需要覆盖由上次操作产生的临时数据,从而会执行大量写前擦除操作。为探究擦除操作对能耗感知模型的影响,实验通过设置不同的内存大小,分别在2GB、4GB、8GB和16GB的情况下依次对5张不同的数据表进行全表扫描并排序,计算不同内存资源下模型对能耗的估算误差。表5展示了数据表信息,实验结果如图4所示。

表5 数据表信息

Tab. 5 Information of data table

表名	记录数	大小/GB
PART	2 000 000	0.23
PARTSUPP	8 000 000	1.13
ORDERS	15 000 000	1.64
LINEITEM1	30 000 000	3.56
LINEITEM2	60 000 000	7.29

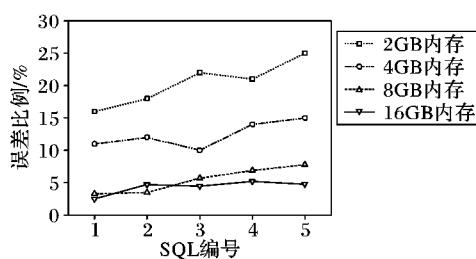


图4 内存大小与模型误差关系

Fig. 4 Relationship of memory size and model error

如图4所示:在内存为2GB时,读操作能耗感知模型的误差较大,平均误差超过20%;当内存分别为4GB和8GB时,平均误差都显著下降,模型误差与内存设置有着明显的关系。仔细观察发现,在以上3种内存设置下,随着SQL语句操作的数据表数据量的增大,模型误差也相应变大,模型误差与SQL语句操作的数据量之间有着明显关系。分析由于内存资源依然紧缺,随着SQL语句产生的临时数据量增大,相

应的擦写临时表空间操作增多,导致模型误差变大;而当内存为16GB时,模型平均误差降到5%,同时模型误差呈稳定趋势。实验结果显示,读操作能耗感知模型在内存资源相对充裕的条件下,准确度更高,也更加稳定。这主要是由于读操作在内存充足的情况下,将产生的临时数据存储于内存空间,减少了对临时表空间的使用,避免了擦除操作对模型的影响。当然,本文模型在竞争环境下的误差高于独占环境的一个主要因素也是由于其他程序和DBMS竞争内存资源导致的。

根据以上实验结果可知,充足的内存资源可以提高读操作能耗感知模型的准确性和稳定性,但对写操作能耗感知模型而言,其擦除操作主要发生在更新操作执行过程中,与内存大小无关,而想要进一步提高写操作能耗感知模型的准确度,只有通过全面地结合CPU指令总数、写操作次数以及擦除操作次数进行线性拟合,从而改进现有模型来实现。由于现有数据库系统主要针对机械磁盘进行设计与优化,磁盘没有SSD写前擦除的特性,因此现有数据库系统无法获知SSD写前擦除的执行次数及功耗等关键信息;同时,擦除操作以数据块为单位,有时即使更新一条记录,也可能会擦除整个数据块。很难根据更新操作的次数来获知擦除操作的执行次数,目前无法改进写操作的能耗感知模型,因此,如何获知擦除操作的执行次数对写操作能耗感知模型进行改进是我们未来的研究重点。

#### 5 结语

本文提出的能耗感知模型通过将建模过程分解为两个相对简单独立的资源开销预测模型(时间开销模型和功耗开销模型),简化了复杂的模型构建过程。模型不需要单独测量系统中各资源部件的能耗,仅需要获取SQL语句执行计划提供的资源消耗信息(CPU指令总数、I/O操作执行次数)就能对DBMS产生的能耗进行有效的估算。目前,在以节能环保为中心的全球低碳化趋势下,研究具有能耗感知的节能数据库系统是学术界和工业界共同关注的热点话题。现有针对基于SSD数据库系统的研究与设计主要以提高性能为目标,缺少针对能耗的感知及处理等方面的研究。本文提出的能耗感知模型,实现了为数据库负载较准确地估算能耗的目的,但是模型在DBMS处于竞争环境下的准确性和稳定性有待提高,以及写操作的能耗感知模型有待改进;同时,需要进一步考虑性能和能耗之间的折中问题。

#### 未来工作要点:

- 1) 研究在竞争环境下( $MPL \geq 2$ ),如何进一步提高能耗感知模型的适应性和健壮性。
- 2) 以本文能耗感知模型为基础,研究如何获取实时系统可接受的性能浮动范围,从而在满足性能需求的条件下进一步降低能耗。
- 3) 研究可以跟踪数据库对SSD执行擦除操作的次数、功耗等信息的方法,从而改进写操作的能耗感知模型,进一步提高模型的准确性和稳定性。

#### 参考文献 (References)

- [1] KATZ R H. TechTitans building boom [J]. IEEE Spectrum, 2009, 46(2): 40–54.



- [2] 吕天文. 2013年数据中心能效现状深度分析[J]. 电源世界, 2013(6): 7–8. (LYU T W. A deep analysis of data center energy efficiency in 2013 [J]. The World of Power Supply, 2013(6): 7–8.)
- [3] 中国IDC圈. 未来五年国内数据中心能耗将翻一番[EB/OL]. (2012-03-29) [2018-01-31]. <http://tech.idcquan.com/pro/34910.shtml>. (China IDC Circle. Energy consumption in domestic data centers will double in the next five years [EB/OL]. (2012-03-29) [2018-01-31]. <http://tech.idcquan.com/pro/34910.shtml>.)
- [4] GLANZ J. Power, pollution and the Internet [N]. The New York Times, 2012-09-22.
- [5] Global Action Plan. An inefficient truth [EB/OL]. [2017-11-01]. <http://globalactionplan.org.uk>.
- [6] POESS M, NAMBIAR R O. Energy cost, the key challenge of today's data centers: a power consumption analysis of TPC-C results [J]. Proceedings of the VLDB Endowment, 2008, 1(2): 1229–1240.
- [7] TSIROGIANNIS D, HARIZOPOULOS S, SHAH M A. Analyzing the energy efficiency of a database server [C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2010: 231–242.
- [8] SCHALL D, HUDLET V. Enhancing energy efficiency of database applications using SSDs [C]// Proceedings of the Third C\* Conference on Computer Science and Software Engineering. New York: ACM, 2010: 1–9.
- [9] 金培权, 邢宝平, 金勇, 等. 能耗感知的绿色数据库研究综述[J]. 计算机应用, 2014, 34(1): 46–53. (JIN P Q, XING B P, JIN Y, et al. Survey on energy-aware green databases [J]. Journal of Computer Applications, 2014, 34(1): 46–53.)
- [10] 王江涛, 赖文豫, 孟小峰. 闪存数据库: 现状、技术与展望[J]. 计算机学报, 2013, 36(8): 1549–1567. (WANG J T, LAI W Y, MENG X F. Flash-based database: studies, techniques and forecasts [J]. Chinese Journal of Computers, 2013, 36(8): 1549–1567.)
- [11] 吕雁飞, 陈学轩, 崔斌. 基于闪存的数据库性能评测与优化分析[J]. 计算机研究与发展, 2009, 46(s2): 682–687. (LYU Y F, CHEN X X, CUI B. Performance evaluation and optimization analysis on flash-based database [J]. Journal of Computer Research and Development, 2009, 46(s2): 682–687.)
- [12] BAUSCH D, PETROV I, BUCHMANN A. On the performance of database query processing algorithms on flash solid state disks [C]// Proceedings of the 2011 International Workshop on Database and Expert Systems Applications. Washington, DC: IEEE Computer Society, 2011: 139–144.
- [13] DO J, PATEL J M. Join processing for flash SSDs: remembering past lessons [C]// Damon 2009: Proceedings of the 2009 International Workshop on Data Management on New Hardware. Providence, Rhode Island: [s. n.], 2009: 1–8.
- [14] PARK S S, LEE S W. Hash join in commercial database with flash memory SSD [C]// Proceedings of the 2010 IEEE International Conference on Computer Science and Information Technology. Piscataway, NJ: IEEE, 2010: 265–268.
- [15] RODRIGUEZ-MARTINEZ M, VALDIVIA H, SEGUEL J, et al. Estimating power/energy consumption in database servers [J]. Pro-  
cedia Computer Science, 2011, 6(1): 112–117.
- [16] XU Z. Building a power-aware database management system [C]// Proceedings of the 2010 SIGMOD PhD Workshop on Innovative Database Research. New York: ACM, 2010: 1–6.
- [17] XU Z, TU Y C, WANG X. Exploring power-performance tradeoffs in database systems [C]// Proceedings of the 2010 IEEE International Conference on Data Engineering. Piscataway, NJ: IEEE, 2010: 485–496.
- [18] XU Z, TU Y C, WANG X. PET: reducing database energy cost via query optimization [J]. Proceedings of the VLDB Endowment, 2013, 5(12): 1954–1957.
- [19] 杨良怀, 朱红燕. 整机系统实时功率剖析与建模[J]. 计算机科学, 2014, 41(9): 32–37. (YANG L H, ZHU H Y. Whole system realtime power profiling and modeling [J]. Computer Science, 2014, 41(9): 32–37.)
- [20] 陈俊, 胡悦, 杨娇, 等. 云计算数据中心实时能耗建模[J]. 计算机工程与设计, 2017, 38(9): 2494–2497. (CHEN J, HU Y, YANG J, et al. Cloud computing data center real-time energy modeling [J]. Computer Engineering and Design, 2017, 38(9): 2494–2497.)
- [21] 国冰磊, 于炯, 廖彬, 等. 结构化查询语言动态功耗解析及建模[J]. 计算机应用, 2015, 35(12): 3362–3367. (GUO B L, YU J, LIAO B, et al. Dynamic power consumption profiling and modeling by structured query language [J]. Journal of Computer Applications, 2015, 35(12): 3362–3367.)
- [22] GUO B L, YU J, LIAO B, et al. A green framework for DBMS based on energy-aware query optimization and energy-efficient query processing [J]. Journal of Network & Computer Applications, 2017, 84(C): 118–130.
- [23] 陆克中, 朱金彬, 李正民, 等. 面向固态硬盘的Spark数据持久化方法设计[J]. 计算机研究与发展, 2017, 54(6): 1381–1390. (LU K Z, ZHU J B, LI Z M, et al. Design of RDD persistence method in Spark for SSDs [J]. Journal of Computer Research and Development, 2017, 54(6): 1381–1390.)

This work is partially supported by the National Natural Science Foundation of China (61462079, 61562078, 61562086), the Science and Technology Support Projects of Ministry of National Science and Technology (2015BAH02F01).

**LI Shu**, born in 1993, M. S. candidate. His research interests include green computing, machine learning.

**YU Jiong**, born in 1964, Ph. D., professor. His research interests include network security, grid computing, distributed computing.

**GUO Binglei**, born in 1991, Ph. D. candidate. Her research interests include green computing.

**PU Yonglin**, born in 1991, Ph. D. candidate. His research interests include green computing, distributed computing.

**YANG Dexian**, born in 1991, Ph. D. candidate. His research interests include green computing.

**LIU Su**, born in 1994, M. S. candidate. Her research interests include distributed computing, in-memory computing.