

引用格式:李中单,严华. 基于数据访问计数的 NAND 闪存缓存管理算法[J]. 微电子学与计算机,2020,37(5):64-68. [LI Z D, YAN H. Cache management algorithm for NAND flash memory based on data access counting [J]. Microelectronics & Computer,2020,37(5):64-68.]

基于数据访问计数的 NAND 闪存缓存管理算法

李中单¹, 严 华²

(1 四川大学 电子信息学院,四川 成都 610065;2 电子信息控制重点实验室,四川 成都 610036)

摘 要:针对现有的 NAND 闪存缓存管理算法对缓冲区替换页选择和替换代价考虑不足,以及算法时间开销大的问题,提出基于数据访问计数的 NAND 闪存缓存管理算法.该算法首先考虑 NAND 闪存读写代价的不同以及数据访问频度的差异,对数据页进行冷干净、冷脏、热干净、热脏划分并分别采用 LRU 队列进行管理.然后,根据提出的数据访问计数策略计算数据页的访问计数值.最后,结合队列长度和数据页访问计数值给出了新的数据页替换策略.基于 QEMU 和 Linux 的仿真实验结果表明,与 LRU、CF-LRU、LRU-WSR、CCF-LRU、LLRU 等现有算法相比,所提算法能够有效降低写闪存次数,减少程序运行时间.

关键词: NAND 闪存;缓存管理;替换页选择;替换代价;访问计数

中图分类号: TP316

文献标识码: A

文章编号: 1000—7180(2020)05—0064—05

Cache management algorithm for NAND flash memory based on data access counting

LI Zhong-dan¹, YAN Hua²

(1 College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China;

2 Science & Technology on Electronic Information Control Laboratory, Chengdu 610036, China)

Abstract: Abstract:In view of the problem that the existing NAND flash cache management algorithm considering insufficiently about the buffer replacement page selection and replacement cost and the algorithm time overhead is large,a NAND flash memory cache management algorithm based on data access counting is proposed. Considering the difference of read and write cost of NAND flash memory and the difference of data access frequency, the proposed algorithm divides data pages into cold clean, cold dirty, hot clean and hot dirty, and manages them by LRU queues. Then, according to the proposed data access count strategy, the access count value of the data page is calculated. Finally, based on queue length and data page access count, a new data page replacement strategy is proposed. The simulation results based on QEMU and Linux show that compared with LRU algorithm, CF-LRU algorithm, LRU-WSR algorithm, CCF-LRU algorithm and LLRU algorithm, the proposed algorithm can effectively reduce the number of writes and reduce the running time of the program.

Key words: NAND flash; cache management; replacement page selection; replacement cost; access counting

1 引言

NAND 闪存有许多特点,例如体积小,重量轻,

速度快,可靠性高,耐震性高等^[1]. 由于这些优点, NAND 闪存如今被嵌入式系统广泛地作为存储介质. 随着存储容量不断增加,以及价格的不断降低,

NAND 闪存正逐渐取代磁盘成为电子设备的主要存储媒介^[2]. NAND 闪存由若干块组成,每个块由一定数量页组成^[3].

由于电子产品其存储空间有限,当缓存区满时,需要采用页面替换算法选择合适的页面进行替换. 针对文献[4-8]所提及算法存在的不足,本文提出基于访问计数的 NAND 闪存缓存管理算法,即 AC-LRU (Access-Counting LRU)算法. 算法按照数据页的冷热区分和读写差异,采用四个 LRU 队列对数据页进行管理;并为每个数据页设置访问计数值,在按照冷干净页、冷脏页、热干净页、热脏页次序选择替换页面时,结合队列长度值,采用新的替换策略来选择替换页面,从而提高命中率和减少读写操作次数. 为了验证算法有效性,实验借助 QEMU 软件建立 Linux 嵌入式仿真平台,从写闪存次数、时间消耗代价等方面与现有算法进行了对比.

2 AC-LRU 算法描述

2.1 队列划分

如图 1 所示,AC-LRU 算法基于页面访问情况以及 NAND 闪存读写代价不一致的特点,把数据页分为四类:冷的干净页、冷的脏页、热的干净页、热的脏页,并根据每类数据页的访问时间,使用四个 LRU 队列进行缓冲区管理,即冷干净队列 CC(Cold Clean LRU list),长度为 L_1 ,冷脏队列 CD(Cold Dirty LRU list),长度为 L_2 ,热干净队列 HC(Hot Clean LRU list),长度为 L_3 ,热脏队列 HD(Hot Dirty LRU list),长度为 L_4 . 设缓冲区大小为 L ,满足 $L=L_1+L_2+L_3+L_4$. 每个队列的尾部数据为对应队列的 LRU 页,在需要进行页面替换时,算法直接对各个队列 LRU 页进行判断和选择.

2.2 冷干净队列最小窗体值设置

根据缓冲区中数据页读写差异以及冷热差异,可将数据页进行区分,不同数据页替换代价有所区别. 现有的缓冲区页面替换代价遵循公式(1)所示:

$$C_{HD} > C_{HC} > C_{CD} > C_{CC}$$
 (1)

式中, C_{HD} 为替换热脏页的代价, C_{HC} 为替换热干净页的代价, C_{CD} 为替换冷脏页的代价, C_{CC} 为替换冷干净页的代价. 通过公式(1)可以看出,在进行缓冲区页面管理时,为减少缓冲区替换代价,应尽可能减少脏页以及热页的替换次数,需要替换的页面应尽可能是代价最小的页面,即冷的干净页面.

由于冷干净页面替换代价最低,现有算法在缓冲区中存在冷干净页时,会优先考虑替换冷干净页.

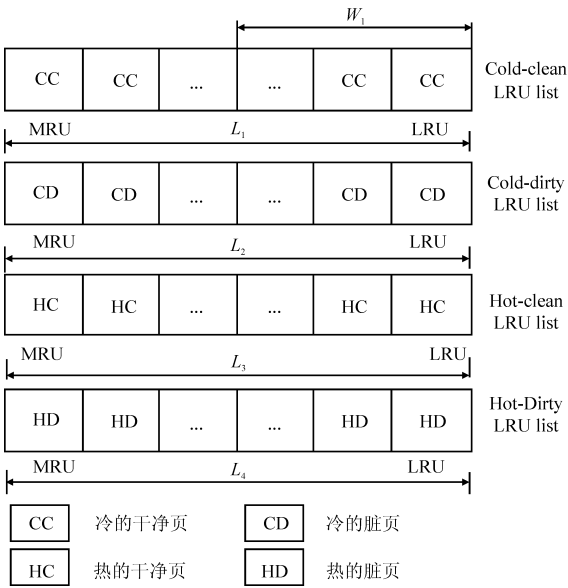


图 1 AC-LRU 算法图解

例如 CCF-LRU 算法在缓冲区中冷干净队列长度值大于 0 时即选择替换冷干净页. LLRU 算法在缓冲区中存在冷干净页时,由于冷干净页命中次数较少且读写代价低则替换值较小,总是会被优先替换. 而实际上很多刚进入缓冲区的冷干净数据页会被再次访问,从而由冷页变成热页,此时算法中的缓冲区数据热度判定对于刚进入缓冲区中的冷干净数据来说是一种热度误判.

AC-LRU 算法根据这种情况,对冷干净队列设置最小窗体值 W_1 ,其值由式(2)可得:

$$W_1 = L * P$$
 (2)

式中, W_1 表示冷干净 LRU 队列最小长度, L 表示缓冲区大小, P 为比例常数.

算法在需要进行页面替换选择时,考虑队列长度是否超过最小窗体值. 若队列长度大于最小窗体值,则替换冷干净队列 LRU 页;若队列长度小于最小窗体值,则替换其他队列 LRU 页. 算法在页面替换选择时根据最小窗体值对冷干净队列长度做限定,使得冷干净队列长度值不至于太小,保证刚进入缓冲区的冷干净页能有停留变为热页的机会,同时也能减少误判概率,提高命中.

2.3 访问计数

由于缓冲区中热数据页会被频繁替换,替换代价也会相应增加. 当前算法在对热数据页考虑时,会尽可能将热数据页保留在缓冲区中. 例如 LRU-WSR 算法在遇到热页时更改其状态为冷页并将其移动至队列首. LLRU 算法考虑热数据页时,采用页面命中次数和读写代价差异值乘积作为替换值,

热页面由于命中次数较多则很难被替换,而实际上由于数据访问局部性,停留在热队列中的页面可能较长时间内没有被再次访问,已经成为“变”冷的热页面,即伪热页。则相对于伪热页,热队列中其余页面为最近经常被访问的真热页。

为了对热页面的热度进行准确判断,AC-LRU算法引入两个访问计数值:全局访问计数值 c_{current} ,以及每个数据页的访问计数值 c_{page} 。 c_{current} 初值为 0,数据页每进行一次读写操作, c_{current} 加 1,然后将当前的 c_{current} 访问计数值赋值给该页面的访问计数值 c_{page} ;若该页被再次访问,则将新的访问计数值赋值给该页,表示该数据页最近一次访问“时间”。

算法考虑页面最近访问“时间”和系统当前访问“时间”,仅对热干净队列和热脏队列中的 LRU 数据页进行判断,判断方法如下所示:

$$p_H = \begin{cases} p_{FH}, & \text{if } c_{\text{current}} - c_{\text{page}} \geq n * L \\ p_{TH}, & \text{if } c_{\text{current}} - c_{\text{page}} < n * L \end{cases} \quad (3)$$

式中, p_H 表示热干净队列和热脏队列中的 LRU 页, p_{FH} 表示页面 p_H 为伪热页, p_{TH} 表示页面 p_H 为真热页。 c_{page} 表示热干净队列和热脏队列中 LRU 页面的访问计数值, c_{current} 表示当前的全局访问计数值, L 表示缓冲区大小, n 为一个常量。当热页面满足式(3)中第一个条件时,则为伪热页面 p_{FH} ;当热页面满足式(3)中第二个条件时,则为真热页面 p_{TH} 。

AC-LRU 算法在进行热数据页的判断时,将热队列 LRU 页访问值与当前访问计数值差值进行判断,通过差值是否超过阈值的方式,能够直观准确判断出页面是否由热“变”冷,从而优先选择替代代价更低的数据页,减少伪热页存在的数量。

2.4 替换策略

在缓冲区满时,需要进行页面替换。现有算法在替换页选择时仅考虑页面替代代价,不结合队列长度进行考虑。CCF-LRU 不考虑冷干净队列长度值变化即选择冷干净页面进行替换。LLRU 算法在替换页选择时,通过命中次数和替代代价乘积选择替换值最小页来调节各队列间长度,而热脏队列 LRU 页由于替代代价大,较晚才能被替换,则热脏队列长度值会越来越大。

而 AC-LRU 算法在进行替换页选择时,综合考虑队列长度和页面替代代价。当冷队列长度值较小时,对长度值较大的热队列 LRU 页进行判断并选择,从而解决队列长度值过大造成算法性能较低问题。

算法按照如下顺序进行替换页选择:

(1)当缓冲区页面未被命中且缓冲区满时,需要进行页面替换。若冷干净队列长度大于最小窗体值 W_1 ,选择冷干净队列 LRU 页面作为替换页;否则,继续后面的步骤。

(2)若冷干净队列长度小于最小窗体值 W_1 ,同时冷脏队列长度大于冷干净队列长度值,且热干净队列 LRU 页根据式(3)判断为伪热页,则选择该页作为替换页;否则,继续后面的步骤。

(3)若冷脏队列长度小于冷干净队列长度值,则判断热脏队列长度是否大于热干净队列长度值。若大于,则根据式(3)判断热脏队列 LRU 页是否为伪热页,如果为伪热页,则选择该页作为替换页;否则,继续后面的步骤。

(4)若以上条件未选出替换页,则选择热干净队列 LRU 作为替换页。

替换页选择策略过程如算法 1 所示。

Algorithm 1

Algorithm SelectVictim-Page

1:Input: CC= cold clean LRU list; CD= cold dirty LRU list;

HC= hot clean LRU list; HD= hot dirty LRU list;

L_1 =size of cold clean LRU list; L_2 =size of cold dirty LRU list;

L_3 =size of hot clean LRU list; L_4 =size of hot dirty LRU list;

W_1 =minimumwindow valueof CC;

2:Output: the victim page for replacement;

3: if $L_1 \geq W_1$

victim page = the LRU page of CC;

4: else

5: if $L_2 \geq L_1$ and the LRU page of HC is old hot clean page
victim page = the LRU page of HC;

6: else if $L_2 < L_1$

7: if $L_4 > L_3$ and the LRU page of HD is old hot dirty page
victim page = the LRU page of HD;

8: else

victim page = the LRU page of HC;

9: returnvictim page;

2.5 算法步骤

在进行页面访问时,按照如下步骤进行页面访问:

(1)当访问的数据页命中且为读操作时,设置相应热标志位且将其移动至热干净队列 MRU 位置。

(2)当访问的数据页命中且为写操作时,设置相应热标志位其将其移动至热脏队列 MRU 位置。

(3)当访问的数据页不命中且缓冲区满时,采用替换策略选择页面进行替换。根据访问操作将页面添加至冷干净队列或者冷脏队列 MRU 位置。

算法 2 描述了 AC-LRU 算法主体部分,包含页面访问过程和替换页选择策略过程.

Algorithm2
Algorithm AC-LRU
1:Input: request for page p , access type is R (read) or W (write)
2:if the hit page in CC or HC
3: if the type is R
 Set hot flag of page p ;
 Move the page to the MRU of HC;
4:else
 Move the page to the MRU of HD;
5:else if the hit page in CD or HD
 Set hot flag of page p ;
 Move the page to the MRU of HD;
6: else if $L_1+L_2+L_3+L_4=L$ and p is not belong to buffer
 Select Victim-Page();
7:if the type is R
 Move the page to the MRU of CC;
8:else
 Move the page to the MRU of CD;

3 实验环境

实验采用 QMENU 搭建嵌入式 Linux 仿真环境,移植 Yaffs2 文件系统,用 NANDSim 来模拟 NNAD 闪存设备. 实验选取 NAND 闪存总容量为 64MB. 实验中所涉及参数为 2. 2 节中窗体值比例常数 p , 2. 3 节中缓冲区倍数常量 n ,对常数 p 取值为 0. 01,对常量 n 取值为 3.

实验中数据读写为按页操作,NAND Flash 的空间利用率为 90%. 如表 1 所示,实验采用 T_1 , T_2 , T_3 三种不同类型的负载. 其中, T_1 为读密集型请求, T_2 为读写均衡型请求, T_3 为写密集型请求,这三种数据请求类型充分显示算法是否能够适应于变化的数据请求. 其中读写比例“90%/10%”表示读写请求分别占总请求的 90%和 10%,分布局部性”80%/20%”表示 80%的请求分布在 20%的数据页上,且满足 Zipf 分布^[9]. 为了保证实验结果客观性,关闭默认的后台垃圾回收,对比算法均采用 FaGC 算法^[10]作为垃圾回收算法.

表 1 数据请求类型			
类型	总请求次数	读写比例	分布局部性
T_1	1 000 000	10%/90%	80%/20%
T_2	1 000 000	50%/50%	80%/20%
T_3	1 000 000	90%/10%	80%/20%

4 实验结果

将 AC-LRU 和 LRU、CF-LRU、LRU-WSR、

CCF-LRU、LLRU 算法进行比较. 对比性能指标为写闪存次数以及程序运行时间.

图 2 为使用不同算法在 1M-5M 缓冲区大小情况下,测试 T_1 、 T_2 和 T_3 数据请求类型时写闪存次数实验对比结果. 由图可知,AC-LRU 算法的写闪存次数均小于 LRU、CFLRU、LRU-WSR、CCF-LRU 和 LLRU. 由表 1 可知,与 T_3 请求类型相比, T_1 和 T_2 数据请求类型中写操作较多. 当缓存容量变大时,由图 2(a)和图 2(b)可知,AC-LRU 的写操作次数明显降低. 这是由于 AC-LRU 算法在选择替换页时,在冷干净队列长度低于最小窗体值从而需要替换冷脏页时,先考虑热干净队列 LRU 页是否为伪热页,从而优先替代代价更低的热干净队列 LRU 页,从而有效地减少了写闪存次数.

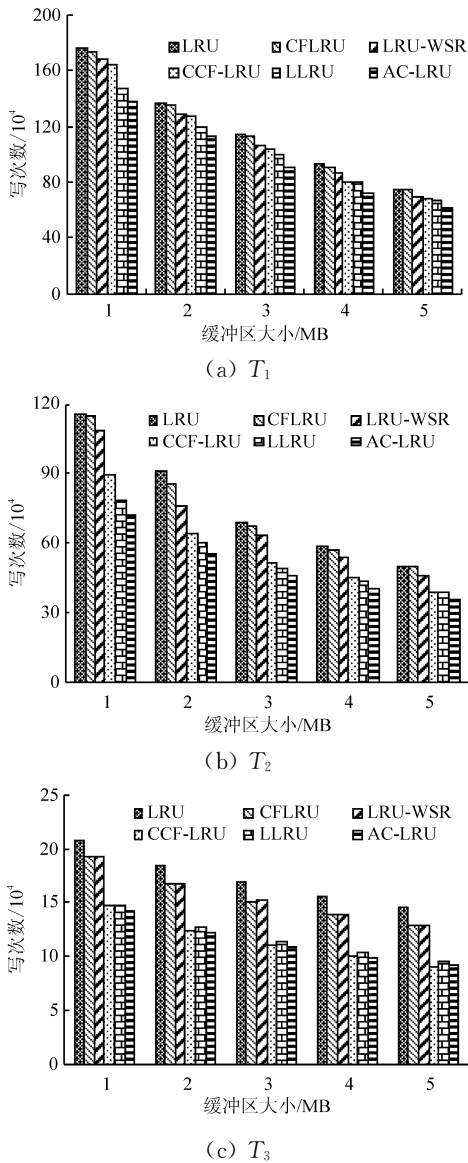
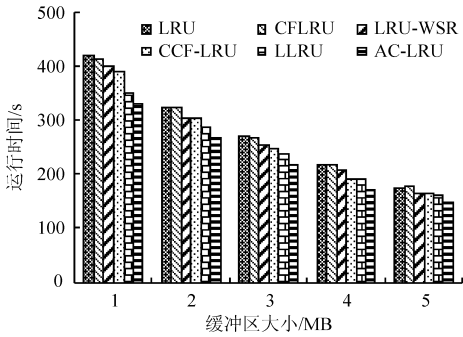


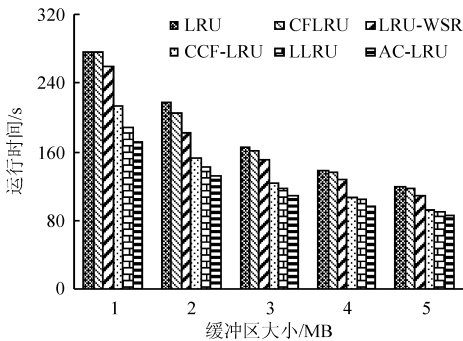
图 2 不同缓冲区大小下写次数对比

另外,在图 2(c)中,由于在 T_3 请求类型中读请求较多,根据 LLRU 算法,热干净页面由于命中次数较多,会被较多的保留在缓冲区中,命中次数较少的脏页会被替换,且缓冲区越大越能显示这种状况.因此在图 2(c)中能看到 LLRU 算法在缓冲区较大时效果较差.

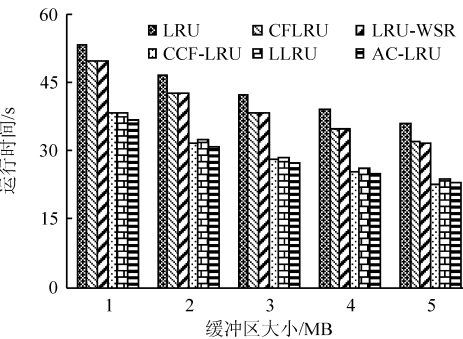
图 3 为使用不同算法在缓冲区变化情况下,运行时间在不同请求数据类型下的对比结果.由图可知,AC-LRU 算法的运行时间均低于 LRU、CFLRU、LRU-WSR、CCF-LRU 和 LLRU.



(a) T_1



(b) T_2



(c) T_3

图 3 不同缓冲区大小下运行时间对比
程序运行时间为程序运行 I/O 时间与 CPU 时间之和, I/O 时间表示为:

$$\text{time} = r \times rc + w \times wc + e \times ec \tag{4}$$

式中, r 为读次数, w 为写次数, e 为擦除次数, rc 为读代价, wc 为写代价, ec 为擦除代价. AC-LRU 算法首先按照冷热以及读写差异对数据页进行划分,把缓冲区分为四个 LRU 队列,在需要进行页面替换时按照替换策略判断队列 LRU 页即可选出替换页,不对热页进行延迟替换,减少了页面移动次数;其次,算法按照新的替换策略,及时替换掉伪热页,减少了读写闪存次数.根据表达式(4)可知,越多的读写操作将会增加更多 I/O 运行时间.因此,AC-LRU 算法运行时间最少.

5 结束语

本文针对基于 NAND 闪存的缓存管理算法的缺陷,提出一种基于数据访问计数的页面替换算法 AC-LRU.所提算法考虑了闪存读写代价差异,同时采用数据访问计数值来进行数据页热度准确区分.按照数据页的读写代价差异和冷热区分把缓存数据页分为四个 LRU 队列,结合数据页的访问计数值和队列长度,采用新的替换策略来进行替换,以减少写操作次数以及运行时间.仿真实验结果显示,AC-LRU 算法在不同读写负载下均可有效减少写闪存的次数,并减少程序运行时间.

参考文献:

[1] CHIANG M L, LEE P C H, CHANG R C. Managing flash memory in personal communication devices [C]// IEEE International Symposium on Consumer Electronics, Singapore,1997, DOI:10.1109/ISCE.1997.658363.

[2] WEI Y, SHIN D. NAND flash storage device performance in Linux file system[C]// International Conference on Computer Sciences & Convergence Information Technology, Kuala Lumpur, Malaysia ,IEEE, 2012.

[3] 于万钧, 张海军, 刘全. 一种基于 NAND 闪存高效的页面替换算法[J]. 现代电子技术, 2017(16):61-64.

[4] 李芳, 徐丽, 陈亮亮. LRU 近似算法的研究[J]. 现代电子技术, 2009, 32(10):36-39.

[5] PARK S Y, JUNG D, KANG J U, et al. CFLRU: a replacement algorithm for flash memory[C]// Cases 06; International Conference on Compilers, Shanghai, China, 2006.

[6] JUNG H, YOON K, SHIM H, et al. LIRS-WSR: Integration of LIRS and Writes Sequence Reordering for Flash Memory[J]. IEEE Transactions on Consumer Electronics, 2008, 54(3):1215-1223.

上,当面临高维数据时,前两种算法运算速度较慢,而 ISMPSO-AKM 与 ADPSO-IKM 算法的收敛趋势大致相同,但在 20~30 次这一迭代阶段,本文提出的算法收敛性能明显提升,说明 ADPSO-IKM 算法可以更快找到全局最优解.在数据集 Breast 上,PSO-KM 受 KM 算法影响较大,呈平稳缓慢的收敛趋势,ADPSO-IKM 算法倾向于在 35 次迭代时稳定收敛,相比之下,ISMPSO-AKM 算法进一步改进聚类效果,聚类性能更好.

5 结束语

本文提出了一种改进的简化均值粒子群 K-means 优化聚类算法(ISMPSO-AKM).将原先简化粒子群的位置项扩展为由个体最优位置、全局最优位置及邻域最优位置线性组合而来的四项.此外,构造了一种基于余弦函数和对数函数的惯性权重.引入 AKM 聚类算法克服了 KM 算法受聚类数及初始聚类中心的选取影响较大的缺点,使聚类结果更准确.对 UCI 数据库中的经典数据集 Iris、Wine、Breast 进行仿真实验,结果表明,与其他改进算法相比,本文提出的 ISMPSO-AKM 算法具体更好的性能.但 AKM 算法的引入,增加了算法整体的运算时间,下一步将对这方面进行研究改进.

参考文献:

[1] 王金永. 基于粒子群优化算法的聚类分析研究[D]. 青岛:青岛理工大学,2015.

[2] 王宏杰,师彦文. 结合初始中心优化和特征加权的 K-Means 聚类算法[J]. 计算机科学,2017,44(11A):467-469.

[3] 李武,赵娇燕,严太山. 基于平均差异度优选初始聚类

中心的改进 K-均值聚类算法[J]. 控制与决策,2017,32(4):759-762.

[4] 陈朝威,常冬霞. 基于密度差分的自动聚类算法[J]. 软件学报,2018,29(4):43-52.

[5] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95-International Conference on Neural Networks. Perth, WA, Australia, IEEE, 2002.

[6] 李斌,王劲松,黄玮. 一种大数据环境下的新聚类算法[J]. 计算机科学,2015,42(12):247-250.

[7] 许明杰,蔚承建,沈航. 基于 Spark 的并行 K-means 算法研究[J]. 微电子学与计算机,2018,35(5):101-105.

[8] 陈小玉,李晓静,马海英. 一种面向大数据的快速自动聚类算法[J]. 计算机应用研究,2017,34(9):2651-2654.

[9] PERAM T, VEERAMACHANENI K, MOHAN C K. Fitness-distance-ratio based particle swarm optimization[C]// Swarm Intelligence Symposium. IEEE, Saint Louis,USA,2012.

[10] 胡旺,李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报,2007,18(4):861-868.

[11] DEEP K, BANSAL J C. Mean particle swarm optimisation for function optimisation[J]. International Journal of Computational Intelligence Studies, 2009, 1(1):72.

作者简介:

靳雁霞 女,(1973-),博士,副教授.研究方向为智能优化算法、虚拟现实技术.

齐欣(通讯作者) 女,(1994-),硕士研究生.研究方向为智能优化算法. E-mail:448463832@qq.com.

张晋瑞 男,(1994-),硕士研究生.研究方向为虚拟仿真.

程琦甫 男,(1994-),硕士研究生.研究方向为虚拟仿真.

(上接第 68 页)

[7] ZHI L, JIN P, XUAN S, et al. CCF-LRU: a new buffer replacement algorithm for flash memory[J]. IEEE Transactions on Consumer Electronics, 2009, 55(3):1351-1359.

[8] 何简繁. 基于 NAND 闪存的固态硬盘缓存优化策略研究[D]. 杭州:杭州电子科技大学,2018

[9] JOHNSON T, SHASHA D, PROPOSED G W. Q: A Low Overhead High Performance Buffer Management Replacement Algorithm[C]// International Conference on Very Large Data Bases. Morgan Kaufmann

Publishers Inc. Athens, Greece, 1994.

[10] YAN H, YAO Q. An efficient file-aware garbage collection algorithm for NAND flash-based consumer electronics[J]. IEEE Transactions on Consumer Electronics, 2014, 60(4):623-627.

作者简介:

李中单 女,(1995-),硕士研究生.研究方向为模式识别与智能系统. E-mail:2412592897@qq.com.

严华 男,(1971-),博士,教授.研究方向为模式识别与智能系统.