

一种 SRAM 辅助新型非易失性缓存的磨损均衡方法

沈凡凡¹⁾ 何炎祥^{1),2)} 张 军¹⁾ 江 南¹⁾ 李清安²⁾ 李建华³⁾

¹⁾(武汉大学计算机学院 武汉 430072)

²⁾(武汉大学软件工程国家重点实验室 武汉 430072)

³⁾(合肥工业大学计算机与信息学院 合肥 230009)

摘 要 随着半导体工艺的发展,处理器集成的片上缓存越来越大,传统存储器面临着存储密度低和漏电功耗高等问题日益严峻.近年来,新型非易失性存储技术展现出漏电功耗低、存储密度高和可扩展性强等优点,是最有潜力构建大容量缓存的新技术.然而,非易失性存储器的写操作次数有限,作为缓存将限制其寿命.同时缓存上的写操作是不均匀的,存在缓存组间和组内的写波动,这一访问特点将导致缓存的每部分磨损不均匀,现有的缓存管理策略却不能感知缓存的写波动.为解决这一问题,提出了 SRAM 辅助新型非易失性缓存的磨损均衡(SRAM-assistEd weAr Leveling, SEAL)方法,该方法包含写波动感知的缓存块迁移算法(Write Variation-aware blOck Migration, WVOM)和阈值指导的缓存块迁移算法(Threshold gUided Block mIgration, TUBI).SEAL 方法重点关注写波动大的缓存组和写强度高的缓存单元,着力减少这部分缓存单元的写压力.WVOM 感知缓存组间写波动并迁移写强度大的缓存组,用于解决组间磨损均衡问题.TUBI 迁移缓存组内写局部性高的缓存块来达到组内磨损均衡.实验结果表明,SEAL 方法与基准配置相比,缓存的磨损程度平均减少了 34.2%,缓存的平均寿命提升了 175%,性能平均提高了 0.735%,系统的动态功耗平均降低了 5.65%.同时,SEAL 方法与最新的研究成果相比,缓存的磨损程度平均减少了 13.1%,缓存的平均寿命提升了 20%.

关键词 非易失性存储器;缓存;磨损均衡;耐久性;写波动

中图法分类号 TP303; TP333

DOI 号 10.11897/SP.J.1016.2017.00743

SRAM-Assisted Wear Leveling for Emerging Non-Volatile Caches

SHEN Fan-Fan¹⁾ HE Yan-Xiang^{1),2)} ZHANG Jun¹⁾ JIANG Nan¹⁾ LI Qing-An²⁾ LI Jian-Hua³⁾

¹⁾(Computer School, Wuhan University, Wuhan 430072)

²⁾(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

³⁾(School of Computer and Information, Hefei University of Technology, Hefei 230009)

Abstract With the development of semiconductor technology and complementary metal oxide semiconductor (CMOS) scaling, the size of on chip cache memory gradually increases in modern processor design. The storage cell density of traditional static random access memory (SRAM) is very low and SRAM has been close to the limit due to its physical property. Furthermore, SRAM consumes a large amount of leakage power which could severely affect the whole system performance. In recent years, the emerging non-volatile memory (NVM) has shown a lot of attractive features, such as low leakage power, high storage density and better scalability. The NVM based cache design has become one of the most promising candidates to build large on chip caches in the

收稿日期:2016-05-11;最终修改稿收到日期:2016-12-26. 本课题得到国家自然科学基金(91118003,61170022,61373039,61402145,61502346,61640220,61662002)、湖北省自然科学基金(2015CFB338)、安徽省自然科学基金(1508085QF138)、江西省教育厅科技项目(GJJ150605)资助. 沈凡凡,男,1987年生,博士研究生,主要研究方向为计算机存储系统和新型非易失性存储器. E-mail: ffshen@whu.edu.cn. 何炎祥(通信作者),男,1952年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为可信软件、分布并行处理和高性能计算. E-mail: yxhe@whu.edu.cn. 张 军,男,1978年生,博士研究生,副教授,中国计算机学会(CCF)会员,主要研究方向为高性能计算和计算机体系结构. 江 南,女,1976年生,博士研究生,讲师,中国计算机学会(CCF)会员,主要研究方向为可信软件和可信编译. 李清安,男,1986年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为编译优化和嵌入式系统. 李建华,男,1985年生,博士,主要研究方向为嵌入式系统、新型非易失性存储器和片上网络.

modern processor architecture. However, both of these NVMs suffer from high write latency and limited write endurance problems. The lifetime of on chip non-volatile caches will be constrained by these issues when we use NVM to architect the cache. What's more, the write operation on cache is unbalanced distribution in the cache set and there exist write variations among inter sets and intra sets. These cache access characteristics will lead to uneven wear for each cache storage cells. Unfortunately, the wear leveling approaches for NVM based main memories could not be simply used to NVM based on chip caches because main memories only have inter set variations. Meanwhile, most of the existing cache management policies are write variation unaware at present. This situation might result in unbalanced write traffic to every cache cells, which causing heavily written cells will fail much earlier than the others. To solve the write endurance problems, this paper proposes a novel technique named SRAM assistEd weAr Leveling (SEAL) for non-volatile caches to minimize both inter and intra set write variations. SEAL contains two algorithms: Write Variation-aware bLock Migration algorithm (WVOM) and Threshold gUided Block mIgration algorithm (TUBI). The main idea of the SEAL scheme focuses on the high write variation cache set and the high write intensive cache storage cells. SEAL attempts to reduce the write pressure of these cache cells. The WVOM algorithm is used to improve inter set wear leveling by detecting the write variation of cache set and then migrating the blocks from write intensive cache sets to the SRAM. The TUBI algorithm aims to migrate the high write locality cache blocks in a cache set to achieve the intra set wear leveling. The experimental results show that, compared with the baseline, on average, the proposed scheme reduces the write variation by 34.2% and thereby improves the wear leveling effectively. In addition, the lifetime of on chip non-volatile caches is significantly improved by 175% and the performance is improved by 0.735%. Furthermore, the dynamic energy consumption of the system is also reduced by 5.65% on average. Meanwhile, The SEAL scheme is compared with the most recent related work. The proposed scheme reduces the write variation by 13.1% on average and the wear leveling is improved simultaneously. Moreover, the lifetime of on chip non-volatile caches is improved by 20% on average.

Keywords non-volatile memory; cache; wear leveling; endurance; write variation

1 引 言

近年来,随着硬件技术的飞速发展,现代集成电路技术要求芯片的集成度高、功耗低和使用寿命长,基于传统硬件搭建的现有计算机存储体系结构面临严峻挑战^[1]. 传统存储技术,包括 SRAM 和 DRAM 技术已经不能适应集成电路技术的新发展. 在这种背景下,新型非易失性存储(Non-Volatile Memory, NVM)技术表现出的优良特性得到了学术界和工业界的广泛关注,例如自旋转移力矩存储器(Spin-Transfer Torque RAM, STT-RAM)^[2]、相变存储器(Phase Change Memory, PCM)^[3]、阻变存储器(Resistive RAM, RRAM)^[4]和赛道存储器(Domain-Wall Memory 或 Racetrack Memory, DWM)^[5]等. 研究者们针对这些新型存储器提出了众多优化技术

来取代传统存储技术. 相比于传统存储技术,这些新型 NVM 具有集成度高、漏电功耗低、非易失性和良好的伸缩性等优点^[6].

然而,由于存储器件的制造工艺和设计原理不同,NVM 的写耐久性(write endurance)远远小于传统存储器. 例如 PCM 的写耐久性约为 10^8 ,它不适用于架构频繁访问的缓存,主要应用于主存^[7-9]. RRAM 的写耐久性有一定提升,但还是在 10^{11} 左右^[7-8,10]. 对于 STT-RAM,虽然有研究表明它的耐久性高达 10^{15} ,但最好的测试结果表明其写耐久性最高仅有 4×10^{12} ^[7-8,11]. 如果再考虑工艺参数变化(process variation),NVM 的写耐久性将更低. 而传统存储技术的写耐久性通常都高于 10^{15} ,不存在存储寿命的问题^[6].

虽然 RRAM 和 STT-RAM 写耐久性可以支持架构缓存,但是应用程序对缓存的访问存在写操作

不均衡的特点,即各个存储单元间的写操作访问存在波动(简称写波动).现有的缓存管理策略均未考虑写操作波动问题,这样会引起CPU频繁地访问少量的存储单元,从而快速损坏这些存储单元,导致整个缓存的寿命下降.由于每个存储单元只能承受一定数量的写操作,因此使用RRAM和STT-RAM架构缓存亟待解决的是存储单元的磨损不均衡问题^[7-8].磨损均衡(wear leveling)技术被广泛用于基于相变存储器的主存中来解决其写耐久性问题^[9,12-14].然而由于缓存与主存的管理方式不同,现有的主存磨损均衡方法不能直接应用在缓存上.CPU对缓存的写操作有组内(intra-set)波动和组间(inter-set)波动两种情况:组内写波动是由于现有缓存替换策略(Least Recently Used,LRU)考虑时间局部性导致少量存储单元被频繁地写入数据;组间波动则是由于应用程序的多样性使CPU对缓存组写操作不均衡.然而,CPU对主存的写操作仅有组间波动的情况^[7-8].同时,目前基于way的混合缓存架构方法无法解决缓存组间写波动问题^[3,11].因此,探索新颖的方法来解决新型非易失性缓存的磨损均衡问题具有十分重要的意义.

针对上述问题,考虑缓存写波动的特征,重点关注写波动大的缓存组和写强度高的缓存单元,着力减少这部分缓存单元的写压力,从而使缓存的磨损更加均衡.本文的创新点在于能周期性检测并感知缓存组间的写波动.如果发现缓存组间写波动过大,则迅速迁移写波动过大的缓存组.同时根据缓存组内的访问特点,通过阈值控制缓存组内的写波动,时时缓解缓存组内写操作不均衡的问题.缓存组内和组间的联合优化可以大幅度降低缓存的磨损.本文的主要贡献总结如下:

(1)提出了一种SRAM辅助新型非易失性缓存的磨损均衡方法(SRAM-assistEd weAr Leveling, SEAL).该方法将写频繁的缓存块迁移到SRAM中;

(2)SEAL方法包含写波动感知的缓存块迁移算法(Write Variation-aware bLock Migration, WVOM)和阈值指导的缓存块迁移算法(Threshold gUided Block mIgration, TUBI).WVOM算法用于解决缓存组间的磨损均衡性,TUBI算法用于解决缓存组内的磨损均衡性;

(3)实验结果表明,相对于最新研究工作,SEAL方法使缓存组间和组内的磨损更均匀,显著的延长了缓存的寿命并降低了系统的功耗;

本文第2节阐述新型非易失性存储技术的背景和磨损均衡方法的相关工作;第3节介绍研究动机;

第4节重点分析并阐述SEAL方法的算法设计和具体实现细节;第5节对实验评估方法和实验结果进行分析与讨论;第6节对全文工作进行总结.

2 背景与相关工作

2.1 非易失性缓存的概述

传统缓存采用SRAM架构,不需要配合内存刷新电路,可提高系统整体的工作效率.但是,它存在漏电功耗大、存储单元面积大及价格相对昂贵等问题.随着移动互联网的兴起,人们对低功耗大容量设备的需求与日俱增.面对这些问题,研究者们探讨了新型非易失性存储技术,如STT-RAM使用磁性隧道结(Magnetic Tunnel Junction, MTJ)存储数据位;RRAM由一个晶体管和一个电阻器组成,利用元器件的电阻转变特性来存储数据;PCM由硫族化合物材料构成的,通过电脉冲来改变该材料结晶状态和非结晶状态,这样可以用来存储数据.表1列出了这些存储技术的重要特征.已有研究表明缓存的功耗占处理器整体功耗的大部分,NVM替代SRAM具有低漏电功耗和高存储密度等优点,能有效的缓解传统存储技术的不足.然而采用NVM架构缓存存在写寿命(耐久性)问题^[3-8].本文专注于讨论使用磨损均衡技术解决新型非易失性存储技术的写耐久性问题.

表1 不同存储器件的特性对比^[3-8]

参数	工艺 制程/nm	特征 尺寸(F ²)	读写 速度	寿命 (耐久性)	漏电 功耗
SRAM	~32	120~200	快	10 ¹⁶	高
STT-RAM	~32	6~50	快/慢	4×10 ¹²	低
RRAM	~11	4~10	快/慢	10 ¹¹	低
PCM	~5	4~12	慢/慢	10 ⁸	低

2.2 NVM磨损均衡的相关工作

为了延长NVM的使用寿命,研究者探讨了减少NVM单元的写操作次数和使NVM各个存储单元的写操作次数尽量均衡的方法.然而,由于应用程序写操作在缓存组间和组内具有分布不均衡的特点,通过减少写操作数量来延长NVM寿命的效果有限.因此,必须引入更加有效的磨损均衡技术使写操作分布更加均衡,提升缓存系统的稳定性.下面探讨磨损均衡技术在主存和缓存上的应用.

2.2.1 基于NVM的主存磨损均衡技术

许多研究者探讨了主存磨损均衡技术,整体的优化目标都是一致的.Qureshi等人^[9]提出Start-Gap磨损均衡方法和随机化Start-Gap方法.Start-Gap

磨损均衡方法是每当主存写操作次数达到阈值时,就交换一次空间上相邻的主存行.这样可以均衡整体写操作,然而它的缺陷在于不能抵抗针对主存行的恶意攻击,随机化 Start-Gap 方法可以解决这类问题. Zhao 等人^[12]提出考虑工艺变化的单层存储单元 (Single Level Cell, SLC) 磨损均衡方法. 该方法动态地将磨损严重且写强度高的多层存储单元 (Multi-Level Cell, MLC) 中的数据迁移到 SLC 存储单元,通过动态阈值控制数据的转换与迁移. 这样充分利用 MLC 高容量的特性并延长其写耐久性. Seong 等人^[13]提出一种动态低开销的磨损均衡方法 Security Refresh. 该方法认为使用 PCM 设计主存时,需要综合考虑存储系统的安全性和耐久性,能抵抗恶意攻击和磨损存储单元. 基于这个思路,该方法提出动态随机化地址映射方法,在每个刷新周期中通过随机密钥交换数据,这保证了数据的安全性,同时也提升了 PCM 的耐久性. Asadinia 等人^[14]提出一种按需的页面配对方法 (On-Demand Page Paired PCM, OD3P). 该方法综合考虑部分主存页面达到其耐久性时,剩余的大多数页面寿命还非常长,而失效页面是 PCM 寿命的决定性因素. 因此,当检测到失效页面时,OD3P 策略将寻找有效页面与其配对,这样可以提升两倍的寿命.

2.2.2 基于 NVM 的缓存磨损均衡技术

近年来一些研究者借鉴主存中的优化方法并结合缓存自身的访问特点,提出了面向缓存的磨损均衡技术. Joo 等人^[15]首先建立了 PCM 缓存的性能、功耗和耐久性模型,然后给出了先读后写和数据位反转及一组磨损均衡方法来提升缓存写耐久性. 该方法扩展自主存中的磨损均衡技术,并未考虑主存与缓存之间管理方式的不同. Wang 等人^[7]提出组内组间写波动感知的缓存管理策略 (Inter/Intra-set Write variation Aware cache Policy, i^2 WAP). 该方法包含两个特征:交换与偏移 (Swap-Shift, SwS) 和概率性缓存行清除 (Probabilistic Set Line Flush, PoLF). SwS 通过周期性交换缓存组的物理地址来减少组间写波动; PoLF 是每当缓存行写命中达到阈值时便清除刚命中的缓存行,这样热数据将被载入到访问次数少的存储单元,从而减少了组内的写波动. 然而, i^2 WAP 的两种方法都存在一定的盲目性,如 SwS 不管缓存组间写波动是否过大都交换相邻的缓存组, PoLF 对于写强度大但写波动小的缓存组,仍然会清除大量缓存块,这样会损失性能. 随后, Jokar 等人^[16]提出使用磨损均衡机制延长 NVM 缓存的寿命. 该方法通过交换缓存组内的写频繁的

缓存行与待替换的缓存行来减少组内磨损. 通过修改写频繁缓存组与写稀少缓存组间的映射关系,交换它们中的数据达到组间磨损均衡. 从缓存组内和组间联合优化提升缓存的整体寿命. 该方法的缓存组内组间交换的思想与 i^2 WAP 的 SwS 方法类似,不同点在于使用静态配置的饱和计数器控制缓存行组内交换和组间映射. 然而该方法因为静态配置的参数,所以不能动态感知缓存组内和组间的写波动,且缓存组间映射逻辑和搜索待交换缓存组的机制会损失系统性能. 本文采用 SRAM-NVM 混合缓存架构能充分利用 SRAM 的高耐久性和写性能,以及 NVM 的非易失性、存储密度高和漏电功耗低等优点,动态迁移缓存组内组间写波动大的缓存行,缓解 NVM 的磨损,能在不损失性能的情况下延长缓存的寿命. Mittal 等研究人员根据缓存组内组内的访问特点,从多个角度探索了延长缓存寿命的方法. Mittal^[17]先提出了用缓存着色的方法减少缓存组间写波动. 该方法通过软件控制主存页面与缓存组间的映射,周期性的修改映射方式,从而确保不同着色缓存上的写操作均衡. 接着, Mittal 等人^[8]提出用于解决缓存组内写波动和延长寿命的均衡写方法 (EqualWrites). 该方法的思想是同一缓存组内两个缓存行的写次数相差大于阈值时,那么就交换这两个缓存行数据. 同时, Mittal 等人^[18]给出了延长基于 way 的混合缓存 (SRAM-NVM) 寿命的方法. 该方法通过数据迁移的方式使 SRAM 优先存储频繁访问且重复使用的数据. 通过减少 NVM 存储单元上的写操作数量达到提升缓存寿命的目的,然而此方法未考虑缓存组内组内的写操作波动的特点. 同时该方法仅考虑了缓存组内的写操作的优化. 而文献[7,16]均表明缓存组间的磨损优化对缓存寿命提升有较大影响. 例如文献[7]的组间磨损优化方法对二级缓存的寿命提升高达 58%,文献[16]的组间磨损优化方法能减少 41% 的组间波动. Lin 等人^[11]提出高耐久性的混合缓存架构,接着给出了缓存分区和访问感知策略,通过写请求的重定向与迁移操作可以有效均衡 STT-RAM 的磨损并延长缓存的寿命.

减少对 STT-RAM 的写操作次数能降低缓存动态功耗和减少缓存的磨损,延长缓存的寿命. Zhou 等人^[19]提出了一种 EWT (Early Write Termination) 的方法来减少 STT-RAM 的写功耗. 当向缓存中写数据时,如果许多相同的数据位已经在缓存中,那么这些写操作是多余的,应该避免这类写操作. 该方法使用 EWT 电路在早期终止重复的 bit-writes 操作,为了判断将要写的位是否重复,采用先读后写

的策略,因为读的代价比写的小,读操作通过电流获取 MTJ 的状态,然后再与将要写的位对比,如果数据相同则不用写入该数据位,这就避免了冗余的写操作. 和 EWT 类似, Bishnoi 等人^[20]提出了一种避免不必要的写操作方法. 该方法可以感知位单元的写操作行为,当位序列第 1 次写时认为是必要的,第 2 次再写同样的序列时就认为是不必要的,这个过程可以由 bit-cell value detection 和 actual write operation 两部分完成:第 1 部分发送一段电流读取当前存储的位单元 MTJ 的值;第 2 部分比较即将写的序列与该值的大小,如果不同则写入数据,反之则避免该写操作,这些都是由写电路完成的. Ahn 等人^[21]提出了一种死写预测的 STT-RAM 缓存架构方法. 大量的数据写入最后一级缓存后再也未使用过,这样的写入操作叫做死写. 该方法首先将死写分为了 3 类: dead-on-arrival、dead-value 和 closing write. 接着通过抽样的方式预测缓存的访问行为,并实时更新预测表,预测表判断缓存的每次写入操作是否是死写,如果是死写,则该写入操作绕过最后一级缓存,直接写入主存或低级缓存. 该方法能减少大部分的死写操作,降低动态功耗,对缓存的寿命有一定的提升作用. 以上方法均能减少写操作的数量,使缓存磨损程度得到降低,进一步优化缓存组内组间的磨损不均衡性将能延长缓存的寿命.

当前研究者使用非易失存储器替代缓存,用于扩大缓存的容量、减少漏电功耗和实现缓存数据的非易失性,而针对优化缓存的磨损均衡性和延长缓存寿命问题的研究仍有很大的提升空间.

3 研究动机

在充分利用 NVM 的低漏电功耗、高存储密度和非易失性等优良特性时,磨损均衡问题也是应用 NVM 架构缓存时必须重点考虑的问题. 因为少数缓存存储单元严重磨损将减少存储系统的可靠性,

更有甚者直接损坏存储子系统,产品的寿命将大大缩短. 磨损均衡技术是延长存储单元寿命的有效手段,它使写操作均衡地分布到所有的存储单元. 写操作的波动特性直接反映出存储单元的磨损均衡的程度. 已有的研究表明,缓存组间和组内存在显著的写波动^[7-8,11,15]. i²WAP 方法通过周期性交换缓存组及清除命中的缓存行来实现磨损均衡^[7],该方法能减少缓存组间和组内的写波动,但是缓存组交换和缓存行清除的阈值设置都是静态配置,不能动态的感知缓存的波动特性,存在盲目性. EqualWrites 方法仅考虑缓存组内的写波动优化^[8],对缓存磨损均衡性提升效果有限. 现有的基于 way 的混合缓存架构(SRAM+STT-RAM)^[11,17,22]通常将写操作压力转移到 SRAM 上,减少 NVM 上的写操作,这在一定程度上能缓解 NVM 内的写波动并减少功耗,但却无法感知缓存组间的写波动.

为了进一步探索缓存组间与组内的写操作压力的分布情况,我们选取 PARSEC 基准测试集^[23]中的应用程序 facesim 为例子进行分析,程序模拟运行 100 亿条指令(10 billion instructions)后统计所有缓存组和缓存行的访问情况,最后一级缓存(Last Level Cache, LLC)的配置为 8 MB, 16 路组相连, 64 字节缓存行大小(详细的实验配置见第 5 节). 图 1 显示了 8192 个缓存组间的写操作分布情况. 很明显写分布不均衡,有 139 个缓存组的写操作超过 6000 次(黑色直线之上的部分),第 7681 个缓存组写次数高达 9289. 同时我们随机选取并查看其中第 1036 个缓存组内的写分布,如图 2 所示,缓存组内的写操作不均衡性也较大,写操作次数波动在 381 次到 1248 之间. 超过平均写次数 562 的有 6 个缓存行(黑色直线之上的部分),如果能有效地将这些波动振幅非常高的写操作转移到 SRAM 上,那么将极大地减小缓存的磨损,缓存行的写寿命(即缓存中写次数最多的缓存行)可以提升 122%((1248-562)/562).

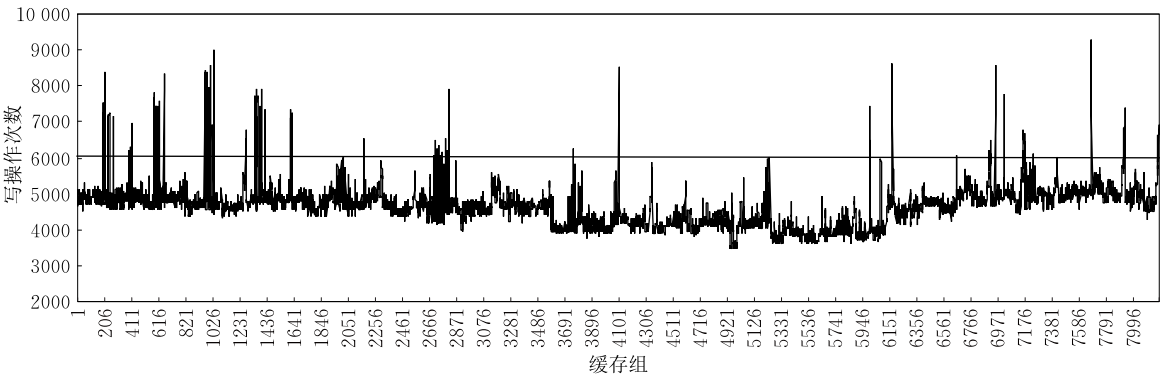


图 1 facesim 测试程序的缓存组间写分布情况(黑色直线为 6000 次写操作)

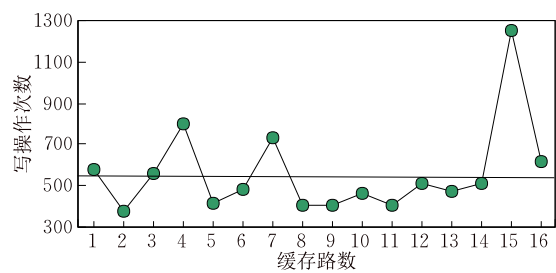


图 2 LRU 替换策略下第 1036 个缓存组内的 16 路写分布情况(黑色直线为 562 次写操作)

综合考虑缓存组间和组内的写操作波动特性,本文引入 SRAM 缓存辅助 NVM 缓存,将 NVM 存储单元中写频繁的缓存块迁移到 SRAM 上. 因为 SRAM 不存在写寿命的问题^[2-3]. 因此可以优化 NVM 的写操作压力,使缓存的磨损更加均衡.

4 SRAM 辅助新型非易失性缓存的磨损均衡方法

本节首先介绍了磨损均衡方法 SEAL 的架构设计和工作原理,接着定义了磨损均衡的评价指标. 其次,为了减少缓存组间写波动,重点描述了写波动感知的缓存块迁移算法(Write Variation-aware block Migration, WVOM). 最后,由于现有缓存替换策略 LRU 会引起 MRU(Most Recently Used)端频繁的访问,给出了缓存组内阈值指导的缓存块迁移算法(Threshold guided Block migration, TUBI).

4.1 SEAL 方法的设计

SEAL 架构采用的是 SRAM 和 NVM 构成的混合缓存. 与基于 way 的混合缓存架构不同,因为此架构无法解决缓存组间写波动. SEAL 架构充分利用 SRAM 的写速度快、无寿命限制和 STT-RAM 的非易失性、存储密度高和漏电功耗低等优点,相互弥补各自的不足. NVM 缓存组及组内写操作频繁的缓存块可以重定向到 SRAM 中,这样可以大幅度缓解和均衡 NVM 缓存组间组内的写压力. 详细的缓存结构如图 3 所示. 该结构包括 SEAL 控制逻辑和混合缓存的组织方式. SEAL 控制逻辑根据缓存的波动情况控制缓存的访问和迁移. 每当访问缓存时,首先判断是否达到预设的缓存组间写波动检测周期 K ,如果达到则执行缓存组间磨损均衡策略,否则执行缓存组内磨损均衡策略(详细的算法设计见 4.3 节和 4.4 节). 混合缓存的组织方式包括 NVM 和 SRAM 两部分,其中 NVM 的每个缓存组

都包含一个缓存组写操作计数器,每个缓存块包含一个额外的标志位 flag(0 表示缓存块在 NVM 中, 1 表示缓存块在 SRAM 中). 当 CPU 读取缓存中的数据时,它根据访问地址中的 flag 标志判断该缓存块所属区域,然后再根据 tag 查找缓存块并读取. CPU 写入数据与读取过程一样,如果写命中 NVM,则缓存组写操作计数器加 1. 当缓存块从 NVM 迁移到 SRAM 中时,需要修改缓存块的 flag 标志,下次访问被迁移的缓存块时将根据 flag 标志判断缓存块所属的位置.

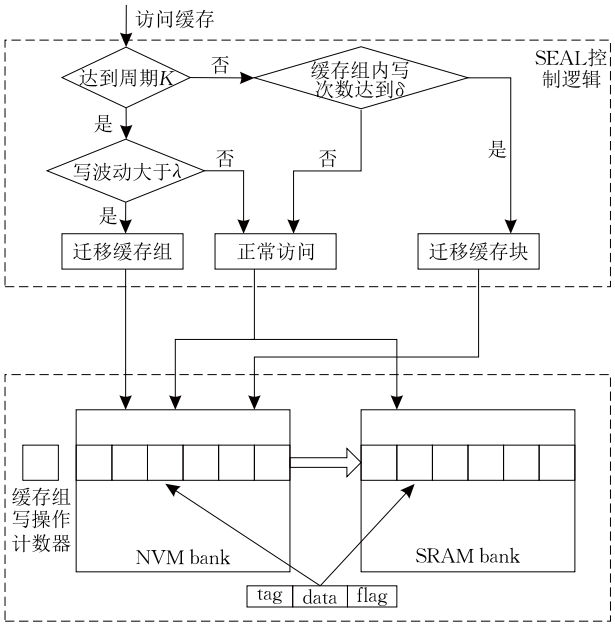


图 3 SEAL 架构下的缓存结构图

图 4 展示了 SEAL 架构下不同时刻缓存块的迁移情况,假设该缓存由 8 组 8 路组相连的 NVM 和 SRAM 构成,接着假设选取 3 个不同时刻 t_1 , K 和 t_n (其中 t_1 和 t_n 为普通时刻, K 为缓存组间写波动检测周期)分别检测 NVM 缓存的访问状态. 当程序运行到 t_1 时刻后,假设缓存组 2 写操作过多,缓存组内存在写波动,那么靠近 MRU 端访问较热的数据块将迁移到 SRAM 中,这样访问较少的冷数据块就有机会载入到 MRU 端的缓存行中. 程序后续的执行过程会重复此步骤,那么,缓存组内的写波动情况会大大缓解,缓存组内的磨损将越来越均衡(详细的算法设计见 4.4 节). 当程序运行到组间写波动检测周期 K 时刻后,如果 NVM 有缓存组(假设组 6)写操作特别多,那么把它们迁移到 SRAM 中将极大缓解缓存组间的写波动. 缓存组 6 的所有行发生迁移(详细的算法设计见 4.3 节). 同理, t_n 时刻对缓存组 4 执行了迁移操作.

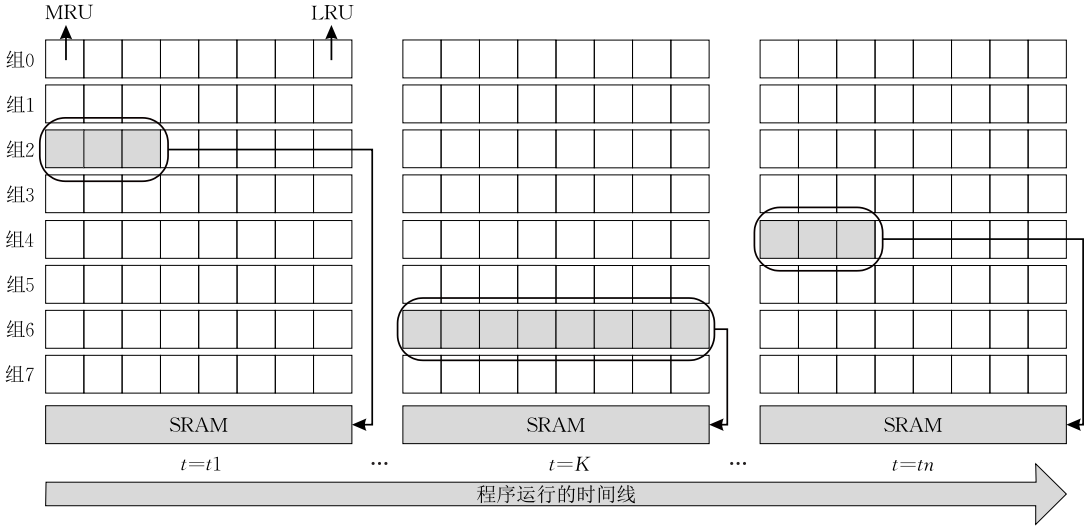


图 4 SEAL 架构下不同时刻的数据迁移情况

4.2 评价指标定义

缓存写操作的波动特性能直接反映出缓存存储单元的磨损均衡程度^[7-8]. 为了评价缓存磨损均衡性, 我们需要量化缓存的写波动, S 表示缓存组的数量, A 表示缓存组的相联度, $w_{i,j}$ 表示第 i 组第 j 路存储单元上写操作的次数, W_{avg} 表示所有存储单元的平均写次数. 这样缓存组间写波动的相关系数 $InterV$ ^[7-8] 可以定义如下:

$$InterV = \frac{1}{W_{avg}} \sqrt{\frac{\sum_{i=1}^S \left(\sum_{j=1}^A w_{i,j} / A - W_{avg} \right)^2}{S-1}}.$$

缓存组内的写波动相关系数 $IntraV$ ^[7-8] 定义如下:

$$IntraV = \frac{1}{W_{avg} \cdot S} \sum_{i=1}^S \sqrt{\frac{\sum_{j=1}^A (w_{i,j} - \sum_{j=1}^A w_{i,j} / A)^2}{A-1}}.$$

以上公式是通过平均的标准方差反映缓存组间和组内写波动特性的. 这两个系数越小, 表示缓存写波动越小, 那么缓存磨损越均衡^[7]. 此评价方法从整体上考虑了所有缓存块上的写操作及平均写操作的情况. 为进一步评价缓存寿命问题, 考虑缓存存储单元磨损最坏的情况, 我们使用缓存中写次数最多的块衡量缓存的寿命. 综合以上两点可以评价本文方法的有效性. 文献^[7-8, 11, 15]均采取了此类评价方法.

4.3 缓存组间磨损均衡策略

4.3.1 WVOM 算法的思想

WVOM 算法的主要思想是: 程序每运行一段时间, 如果缓存组间写波动过大, 那么 WVOM 算法

就集中精力减少写操作非常大的缓存组; 如果缓存组间波动不大, 则正常访问缓存. 具体表现如下: WVOM 算法周期性检测缓存组间的写波动 $InterV$, 当它大于波动阈值 λ 时, 选择写操作次数最多的几个缓存组, 然后将其中的缓存块迁移到 SRAM 中, 那么这部分缓存组的写压力被转移到了 SRAM. 在程序后续运行周期中, 相比其它缓存组, 写操作过多的缓存组中的数据都会被迁移. 这样就会使缓存组间的写操作更加均衡, 同时由于一个缓存组中的写操作总量大幅度减少, 那么分布到组内每个缓存行上的写操作数量也会相应减小. 因此, 缓存的寿命也得到了提升. 假设将图 1 中缓存组写操作超过 6000 次的 139 个缓存组转移到 SRAM 中, 那么整体写分布更均匀.

4.3.2 WVOM 算法的设计

WVOM 算法的伪代码表述形式如算法 1 所示. 每当程序运行 K 个时钟周期时, WVOM 算法感知缓存组间的写波动 $InterV$ 并计算一次, 然后判断其是否在预先设置的阈值 λ 范围内 (第 1 行到第 2 行), 这一过程可以减小算法的计算开销. 如果 $InterV$ 超过 λ 就执行缓存组间的磨损均衡机制, WVOM 从缓存组写操作计数器 $setWrites_n$ 中选择写次数最多的 αN ($0 < \alpha < 1$, N 为缓存组数量) 个缓存组, 并记录这些缓存组号 $sids$ (第 4 行), 接着将 $sids$ 中对应缓存组的所有缓存块都迁移到 SRAM 中, SRAM 使用 LRU 策略替换缓存块, 这一过程结束后便将已经迁移的缓存块状态设置为无效, 下次访问这些缓存行时将被引导到 SRAM 中 (第 5 行到第 7 行). 由于缓存块发生迁移的间隔周期 K 较长, 磨损均衡机制带来的算法开销较小. 最后将所有记录缓存组

写操作的计数器 $setWrites_n$ 值减半,这样做既保留了缓存组当前的访问行为又能减少计数器的存储开销(第 8 行).如果 $InterV$ 小于 λ ,则说明上一周期迁移效果显著,缓存组间写波动较小,不需要执行数据迁移操作,正常访问缓存即可(第 10 行).

算法 1. 写波动感知的缓存块迁移算法.

输入: K : 写波动检测周期;
 λ : $InterV$ 的波动阈值;
 α : 缓存组波动系数;
 N : 缓存组的数量

```
1. FOR 每个周期  $K$  DO
2.   IF  $InterV > \lambda$  THEN
3.     /* 执行缓存块迁移操作 */
4.     从  $setWrites_n$  中选择写操作次数最大的  $\alpha N$  个
       缓存组,并记录组号  $sids$ ;
5.     根据  $sids$  将这些缓存组中的缓存块迁移到
       SRAM 中;
6.     使用 LRU 策略替换 SRAM 中的缓存块;
7.     根据  $sids$  将这些缓存组中的缓存块设置为无效;
8.     将  $setWrites_n$  的值减半;
9.   ELSE
10.    正常执行缓存读写访问操作;
11.  END IF
12. END FOR
```

4.4 缓存组内磨损均衡策略

4.4.1 TUBI 算法的思想

WVOM 算法可以减少写强度高的缓存组中写操作次数,能促进这些缓存组内写操作总数的减少,分布在存储单元上的写操作次数也会减少,从而降低了缓存组内磨损的程度.为进一步优化缓存组内的磨损,TUBI 算法用于解决缓存组内磨损均衡问题,主要思想是:当前大多数处理器的缓存管理策略均采用 LRU 替换算法,该算法考虑缓存访问的时间局部性特点,却没考虑到少量的缓存块可能被频繁写(靠近 MRU 端),这引起了磨损不均衡现象,MRU 端的存储单元磨损速度加快会降低整个缓存的寿命,TUBI 着力解决缓存组内写频繁的热数据.当某个缓存组写操作过多时,便迁移少量局部性高的缓存块到 SRAM 中,这样其它冷数据有机会载入到这些访问热的区域,从而达到磨损均衡的目的.这也有利于促使缓存组内波动变小. TUBI 算法能减少缓存组内的热数据,间接的降低了该缓存组变为高写强度缓存组的概率,缓存组的迁移次数将减少.

4.4.2 TUBI 算法的设计

TUBI 算法的伪代码表述形式如算法 2 所示.对于 NVM 缓存组的每一次写操作,TUBI 将检查

该缓存组写操作总数 $setWrites_n$ 是否达到阈值 δ ,如果达到则将该缓存组内靠近 MRU 端的 φ 个缓存块迁移到 SRAM 中(第 1 行到第 4 行).SRAM 中执行 LRU 替换策略,同时将 NVM 中已经迁移的 φ 个缓存块状态设置为无效. CPU 下次访问这些缓存块时,将在 SRAM 中通过缓存的 tag 查找对应的缓存块,最后再执行写操作(第 5 行到第 7 行). TUBI 时时监测所有缓存组内写操作的变化,这保证了算法的时效性,充分利用缓存的局部性,接下来对热数据的写操作转移到了 SRAM,冷数据载入热行中,那么磨损更均衡.每次执行写操作前仅需要一个比较器硬件电路,用于比较 NVM 缓存组内写操作次数与 δ 的大小,判断组内写操作次数是否达到阈值,比较器电路开销较小^[8]. 如果 $setWrites_n$ 没有达到阈值,则执行正常的缓存写操作.

算法 2. 阈值指导的缓存块迁移算法.

输入: δ : 控制缓存组内写数量的阈值;
 φ : 缓存组内待迁移的缓存块数量

```
1. FOR 每个组  $n$  的写操作 DO
2.   IF  $setWrites_n \% \delta == 0$  THEN
3.     /* NVM 缓存组内写次数到达  $\delta$  时,便迁移该
       组内 MRU 端的  $\varphi$  个缓存块到 SRAM 中 */
4.     将当前组中  $\varphi$  个写频繁的缓存块迁移到
       SRAM 中;
5.     使用 LRU 策略替换 SRAM 中的  $\varphi$  个缓存块;
6.     将 NVM 中的  $\varphi$  个缓存块设置为无效;
7.     执行写操作;
8.   ELSE
9.     正常执行写操作;
10.  END IF
11. END FOR
```

5 实验评估

本部分首先介绍了实验环境配置、SEAL 方法的实现细节和基准测试程序集的特性.随后通过对实验结果的分析验证了本文所提出的 SEAL 方法的有效性.

5.1 实验环境

本文的测试平台采用了 GEM5^[24] 模拟器. GEM5 可以按照指令周期模拟计算机硬件系统运行状况.表 2 展示了具体的实验参数配置.包含四核处理器、三级 cache 结构和 4 GB 的主存.其中共享三级缓存为混合缓存架构(SRAM+RRAM),NVM 以 RRAM 为例子,同时以 SRAM 辅助磨损均衡.基

准配置为不做磨损均衡处理. 混合缓存的参数配置是从修改的 CACTI^[25] 和 NVSim^[26] 中获取. 我们探索了 SEAL 方法所使用参数的多组配置, 最合适的配置为 K 为一千万个周期 (10 million cycles), α 为 2%, N 为 8192 个缓存组, λ 为 10%, δ 为 16, φ 为 3, 详细参数选取分析见 5.3.3 节.

表 2 模拟参数配置	
参数	配置
处理器	4 核乱序执行处理器, 频率为 3GHz, alpha 架构
一级缓存	私有缓存, 指令数据缓存为 32 KB, 8 路组相连, 缓存块大小为 64 B, LRU, 读写为 2 个周期
二级缓存	私有缓存, 缓存大小为 256 KB, 8 路组相连, 缓存块大小为 64 B, LRU, 读写为 8 个周期
混合三级缓存	共享缓存, 1 个 RRAM bank 和 1 个 SRAM bank
RRAM bank	缓存大小为 8 MB, 16 路组相连, 缓存块大小为 64 B, LRU, 读写为 15/66 个周期, 读写功耗为 0.58/0.93 nJ
SRAM bank	缓存大小为 256 KB, 16 路组相连, 缓存块大小为 64 B, LRU, 读写为 15 个周期, 读写功耗为 0.61 nJ
主存	大小为 4GB, 读写为 200 个周期

SEAL 方法是通过修改模拟器的代码实现的. 主要修改了缓存体系结构, 将三级缓存修改为 2 个 bank 结构. 在缓存实现模块中加入所提出的磨损均衡算法, 控制缓存块的访问和迁移操作. 每个缓存组增加了 40 位来记录 $setWrites_n$ 计数器的值, 如果某个缓存组发生写操作则对应的计数器加 1. 算法中的其它参数均为常量, 开销非常小. GEM5 中缓存替换策略 LRU 是使用软件方法实现的, 而在硬件上是通过维护 age 计数器实现的, 那么在实现缓存组内磨损均衡策略时, 将通过 age 计数器来判断 MRU 端的缓存块.

测试程序集选取由多线程应用程序组成的 PARSEC^[23], 所有测试程序的特点如表 3 所示, 它们包含金融分析、计算机视觉和工程应用等多个领域. 测试程序的输入集选取 *simlarge*, 为了有效评价实验效果, 所有程序都快速运行到 ROI (Region Of

表 3 测试程序集的特点		
测试程序	应用领域	数据共享
blackscholes	金融分析	低
bodytrack	计算机视觉	高
canneal	工程应用	高
dedup	企业存储	高
facesim	动画	低
ferret	相似性搜索	高
fluidanimate	动画	低
freqmine	数据挖掘	高
streamcluster	数据挖掘	低
swaptions	金融分析	低
vips	图像处理	低
x264	图像处理	高

Interest), 接着预热 1 亿 (100 million) 条指令, 最后每个程序运行 100 亿 (10 billion) 条指令.

5.2 实验结果

本节详细对比和分析了磨损均衡性的两个重要评价指标: 缓存组内组间的写波动系数和缓存的使用寿命.

5.2.1 磨损均衡性评估

磨损均衡性主要通过缓存组内和组间的写波动相关系数 $IntraV$ 和 $InterV$ 来度量的. 为了评价 SEAL 方法的有效性, 我们对比了研究方法 AYUSH^[18]、 i^2WAP ^[7] 和基准配置. 由于文献 [16] 中的方法和 i^2WAP 类似, 并且文献 [16] 中的方法修改硬件映射逻辑和搜索待交换缓存组的开销较大, 故本文选取 i^2WAP 进行实验对比. AYUSH 方法旨在提升 SRAM-NVM 混合缓存的寿命, 减少缓存组内 NVM 上的写操作, 未优化缓存组间写波动. 由于 i^2WAP 的 SwS 方法需要运行 1000 亿条指令以上, 缓存组间地址映射和交换的轮数才会越多, 从而达到组间磨损均衡的目的. 然而当指令执行数量不够时, 效果不够显著. 因此, 本文修改了 i^2WAP 方法, 将 WVOM 和 PoLF 方法结合, 简称 M- i^2WAP (Modified i^2WAP), 并与其进行对比.

图 5 中展示了缓存组内组间写波动的分布情况. 由图中可知, SEAL 方法的效果均优于 AYUSH、M- i^2WAP 和基准配置, 从整体上看, 写波动平均减少了 13.1%、8.5% 和 34.2%. 从缓存组内看, 组内写波动平均减少了 5.2%、8.5% 和 25.2%. 从缓存组间看, SEAL 方法相对于基准配置减少了 9%. 这说明, SEAL 方法使存储单元上的写操作分布更加均匀, 各个存储单元的磨损程度更小, 进而在整体上提升了 NVM 写操作的耐久性, 延长了缓存的寿命. 例如: facesim 程序的组内写波动很小, SEAL 方法对其影响较小, 但是它的组间波动很大, SEAL 方法能减少一倍以上的写波动. 然而 AYUSH 方法对 facesim 程序的缓存组间写波动影响较小, 几乎没有减少缓存组间写波动. 同样, 对于 canneal 和 ferret 应用程序, AYUSH 方法减少缓存组间波动也较细微. 因为 AYUSH 方法在缓存组内迁移写操作, 并不能感知缓存组间的写波动. SEAL 方法能同时调节缓存组间和组内的写波动. 可见 SEAL 方法针对不同写强度的应用程序具有很强的适应性.

SEAL 方法的有效性表现在, 能根据程序运行状态动态地感知缓存组中的写波动情况, 将波动较大的缓存组中的数据迁移到 SRAM 中, 这样一方面

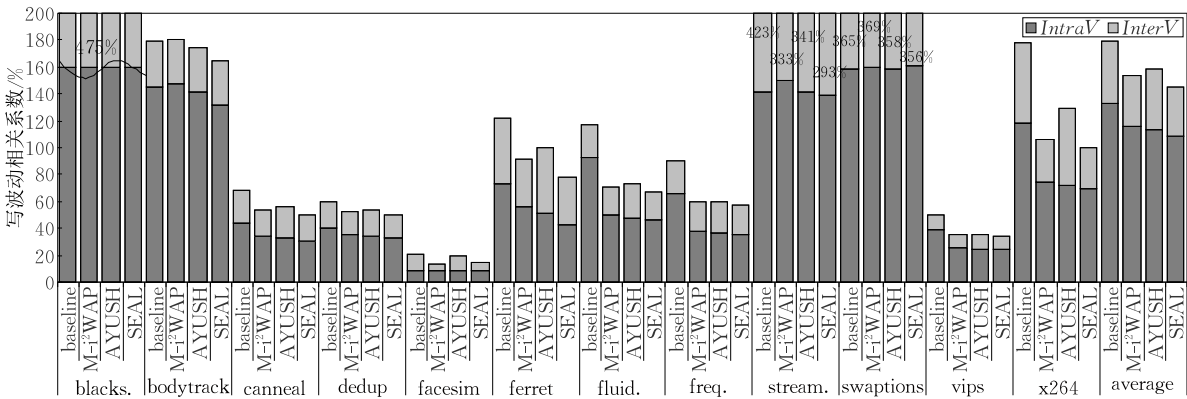


图 5 SEAL 方法与 AYUSH、M-i²WAP 及基准配置的组内组间写波动对比情况

缓解了 NVM 的写压力,另一方面也提升了系统的性能,因为 NVM 缓存中可以存放更多数据,同时 SRAM 的写性能优于 NVM, i²WAP 方法的局限性在于不能检测程序运行过程中的写波动,同时周期性清除命中的缓存行(flush cache line)存在一定的盲目性,且会损失系统性能. AYUSH 方法仅专注于解决缓存组内写压力,对缓存组间波动影响较小,故缓存的磨损均衡性提升效果有限.

5.2.2 缓存的寿命评估

通过减少缓存组内和组间的写波动在一定程度上能促进缓存寿命的提升. 图 6 显示了缓存寿命归一化后的提升情况. 从整体上看, SEAL、AYUSH 和 M-i²WAP 方法均能提升缓存的寿命,且效果显著. 平均寿命提升为 175%、155% 和 159%. 其中, dedup 应用程序寿命提升最明显,因为 SEAL 方法能捕捉到 dedup 中写次数最大的缓存行并将其迁移

到 SRAM,而 facesim 程序虽然其写强度非常大,但是其组内和组间写波动均小于 dedup,故其寿命提升相应少点. 对于 blackscholes 和 swaptions 应用程序,它们的写波动都非常大,但是寿命却几乎没有提升,通过分析实验数据,我们发现这两个应用程序的写强度非常低,绝大多数缓存组写操作小于 10 次,这样发生迁移的缓存块数量少,因此, SEAL 方法发挥的作用也小. AYUSH 方法能提升缓存的寿命,但其提升效果均小于 SEAL 和 M-i²WAP 方法,可见缓存组间写波动对缓存的整体寿命是有影响的. 例如对于 facesim 程序, SEAL 和 M-i²WAP 方法均减少了缓存组间写波动,相应的缓存寿命提升效果均高于 AYUSH. 相比于 AYUSH 和 M-i²WAP 方法, SEAL 方法均优于它们,这说明 SRAM 辅助 NVM 的效果更显著,能转移 NVM 的磨损压力,从缓存组内和组间减少 NVM 上的写操作,从而延长 NVM 的寿命.

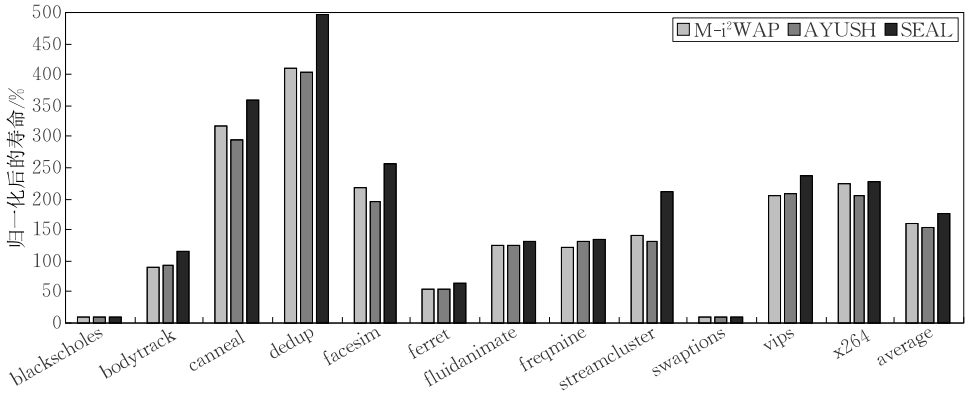


图 6 缓存寿命归一化后的提升情况

5.3 讨论与分析

本节从系统功耗、系统性能、实验参数选取和系统开销等方面进行了讨论.

5.3.1 功耗评估

和传统 SRAM 架构三级缓存相比, NVM 缓存

的静态功耗非常低,混合三级缓存架构将大幅度减少静态功耗. 由于本文的方法主要对动态功耗产生影响,下面重点分析动态功耗,包括缓存块迁移后带来的访问功耗收益(因为 SRAM 写功耗小于 RRAM)和额外功耗开销. 由于 i²WAP 未涉及功耗

评估,因此 SEAL 方法将和基准配置对比.

图 7 展示了归一化后的动态功耗对比情况. 可以看出,除了 blackscholes 和 swaptions 程序几乎没有减少外,其余程序均有相应的减少. SEAL 方法的功耗比基准配置平均减少 5.65%. NVM 中被迁移的缓存块写功耗的减少促进了整体动态功耗的减少. 缓存块迁移引起的功耗开销为一次 NVM 读功耗和一次 SRAM 写功耗. 当然,SEAL 方法也会引入额外的功耗开销,但是由于写波动检测周期较长,约 1000 万个周期(10 million cycles),因此针对标记和计数器的访问功耗开销也比较小,可以忽略不计. 因此,综合以上分析,系统的整体功耗降低了.

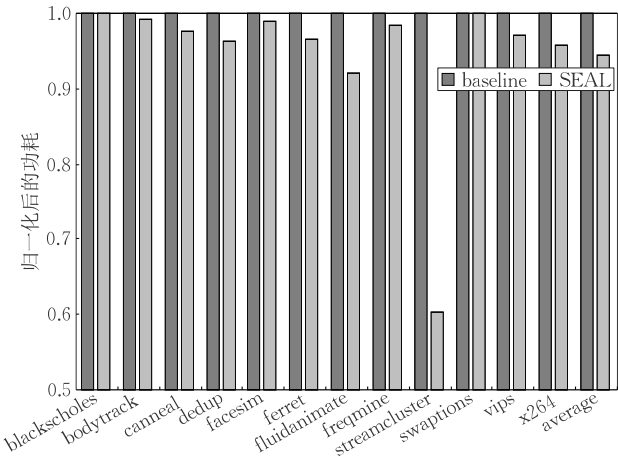


图 7 归一化后的功耗情况

5.3.2 性能分析

图 8 显示了归一化后系统性能(Speedup)的对比情况,性能是由 IPC(Instruction Per Cycle)衡量的. 从平均上看,M-i²WAP 方法相对于基准配置损失了 0.268%的性能,SEAL 方法相对于基准配置提升了 0.735%的性能. 这是因为 M-i²WAP 会周期

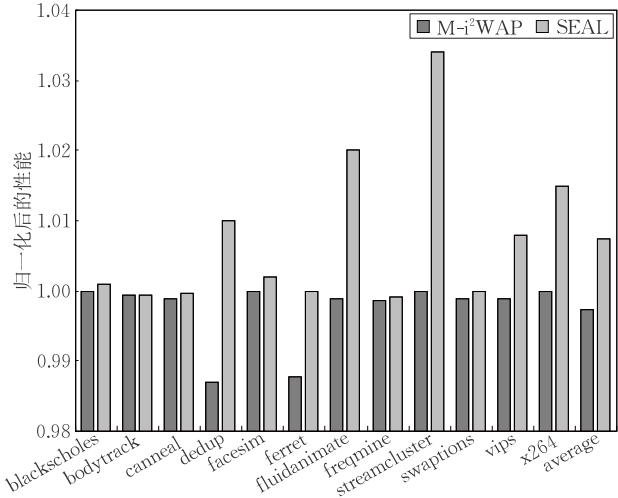


图 8 归一化后的性能情况

性地清除命中的缓存行,导致缓存的命中率下降,从而损失性能. 从图 7 的实验结果可知,SEAL 算法带来的性能开销较小,因为写波动检测周期较长,缓存块迁移不频繁. 同时 SEAL 方法转移了部分写压力,且 SRAM 的写性能优于 NVM,因而能获得性能提升.

5.3.3 参数选取分析

为了发挥 SEAL 方法最大的效果,需要选取合适的参数. 我们测试了参数的多种组合情况,由于测试情形较多,现在仅列出 6 组(G1 和 G2 等)代表性的参数配置组合,如表 4 所示. 图 9 显示了对应的参数组合下性能对比情况,从整体的平均性能上看,G3 是比较合适的参数配置,其它组合均有略微的性能损失. 因为针对 WVOM 算法,综合考虑系统开销和算法效果,选取 K 为一千万个周期(10 million cycles)时的效果和开销均衡,因为当周期太短时,算法开销过大,周期太长时,算法感知缓存访问的局部性就不够. 缓存组间波动阈值 λ 用于帮助减少算法的开销,因为当应用程序的写波动极小时,算法可以不执行,故 λ 取为 10%. α 用于选取多少个写次数最多的缓存组发生迁移,通过实验数据分析,在每个周期里面只有少量的缓存组写次数过多,而这些缓存组中的缓存块对缓存的寿命和波动影响较大,当 α 取 2%时,已经足够获取写操作最多的组了. 针对 TUBI 算法,测试了多组参数配置信息, δ 为 16, φ 为 3 时,能保证迁移数据再被访问的局部性,并可避免缓存组内写操作过大. 以上参数的选取是根据 PARSEC 测试集测试得出,对于其它类型的测试程序,如大数据应用程序 BigdataBench 等,参数的选取需要根据相应的测试集进行优化.

表 4 6 组参数配置

	K/M	$\lambda/\%$	$\alpha/\%$	δ	φ
G1	5	5	2.0	16	3
G2	5	10	2.0	16	3
G3	10	10	2.0	16	3
G4	10	10	2.5	16	3
G5	15	15	2.5	24	3
G6	15	15	2.5	24	4

5.3.4 硬件开销分析

SEAL 方法的硬件开销主要包括额外的 SRAM 存储空间、记录缓存组的写次数 $setWrites_n$ 、参数阈值和两个比较器电路. 前两项为主要开销,SRAM 存储空间开销为 3.1%(256 KB/8192 KB),当然如果想进一步提升系统性能和减少 NVM 磨损,可以

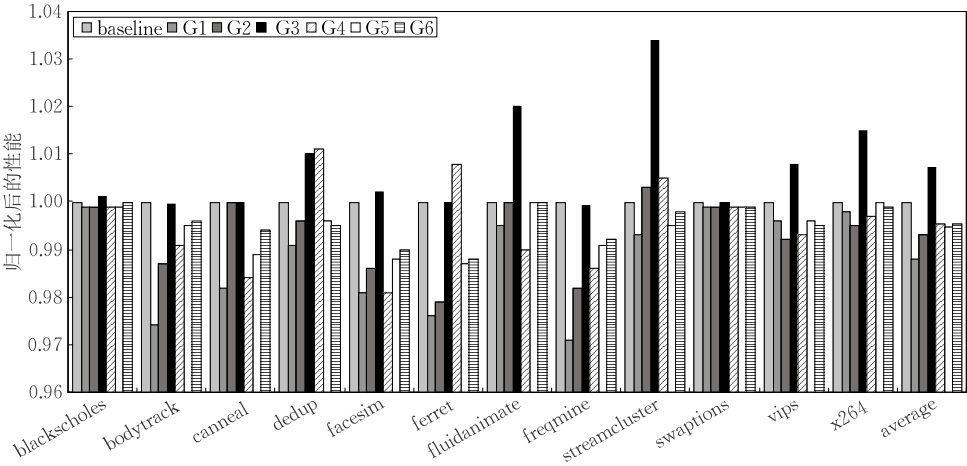


图 9 不同参数配置下的归一化性能对比

适当增加 SRAM 的存储空间. 每个缓存组的计数器 $setWrites_n$ 使用 40 位存储空间, 则开销为 0.49%. 参数阈值为固定值, 所需要位数少, 可以忽略不计. 综上所述, SEAL 方法在可接收的开销范围内能有效均衡缓存的磨损情况并延长缓存的寿命.

6 结束语

NVM 是最有潜力取代 SRAM 缓存架构的下一代存储技术. 具有高可扩展性、低功耗和高存储密度等优点, 然而 NVM 的存储单元写操作次数有限, 且已有的缓存管理策略不能感知写操作波动. 针对这些问题, 本文提出了一种 SRAM 辅助新型非易失性缓存的磨损均衡方法. 该方法使用 WVOM 算法感知缓存组间波动并迁移写强度大的缓存组, 使用 TUBI 算法迁移缓存组内写频繁的缓存块来减小组内写波动. 模拟仿真实验效果表明, 该方法使写操作分布比现有磨损均衡方法更加均匀, 达到了更好的磨损均衡效果. 同时, 缓存的寿命有较大幅度的提升, 系统的功耗也相应的降低了, 性能有略微的改善, 额外的硬件开销也在可以接受的范围内.

致 谢 感谢对本文工作给予支持和建议的匿名评委. 特别感谢武汉大学计算机学院何炎祥教授主持的博士讨论组上的同学和老师提出的宝贵意见和建议!

参 考 文 献

[1] Shen Zhi-Rong, Xue Wei, Shu Ji-Wu. Research on new non-volatile storage. Journal of Computer Research and

Development, 2014, 51(2): 445-453(in Chinese)
(沈志荣, 薛巍, 舒继武. 新型非易失存储研究. 计算机研究与发展, 2014, 51(2): 445-453)
[2] Chang M, Rosenfeld P, Lu S, et al. Technology comparison for large last-level caches (L3Cs): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM// Proceedings of the 19th International Symposium on High Performance Computer Architecture (HPCA2013). Shenzhen, China, 2013: 143-154
[3] Wu Xiao-Xia, Li Jian, Zhang Li-Xin, et al. Hybrid cache architecture with disparate memory technologies//Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA 2009). New York, USA, 2009: 34-45
[4] Li Yong, Chen Yi-Ran, Jones A. A software approach for combating asymmetries of non-volatile memories//Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED 2012). New York, USA, 2012: 191-196
[5] Venkatesan R, Kozhikkottu V, Augustine C, et al. TapeCache: A high density, energy efficient cache based on domain wall memory//Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design. New York, USA, 2012: 185-190
[6] Mao Wei, Liu Jing-Ning, Tong Wei, et al. A review of storage technology research based on phase change memory. Chinese Journal of Computers, 2015, 38(5): 944-960 (in Chinese)
(冒伟, 刘景宁, 童薇等. 基于相变存储器的存储技术研究综述. 计算机学报, 2015, 38(5): 944-960)
[7] Wang Jue, Dong Xiang-Yu, Xie Yuan, et al. i²WAP: Improving non-volatile cache lifetime by reducing inter-and intra-set write variations//Proceedings of the 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA2013). Shenzhen, China, 2013: 234-245
[8] Mittal S, Vetter J S. EqualWrites: Reducing intra-set write variations for enhancing lifetime of non-volatile caches. IEEE Transactions on Very Large Scale Integration (VLSI) Systems,

2015, 24(1): 103-114

[9] Qureshi M K, Karidis J, Franceschini M, et al. Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling//Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture. New York, USA, 2009: 14-23

[10] Kim Y B, Lee S R, Lee D, et al. Bi-layered RRAM with unlimited endurance and extremely uniform switching//Proceedings of the 2011 Symposium on VLSI Technology (VLSIT). Honolulu, USA, 2011: 52-53

[11] Lin I C, Chiou J N. High-endurance hybrid cache design in CMP architecture with cache partitioning and access-aware policies. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2015, 23(10): 2149-2161

[12] Zhao Meng-Yin, Jiang Lei, Zhang You-Tao, et al. SLC-enabled wear leveling for MLC PCM considering process variation//Proceedings of the 51st Annual Design Automation Conference (DAC). New York, USA, 2014: 1-6

[13] Seong N H, Woo D H, Lee H H S. Security refresh: Prevent malicious wear-out and increase durability for phase-change memory with dynamically randomized address mapping//Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA). New York, USA, 2010: 383-394

[14] Asadina M, Arjomand M, Azad H S. Prolonging lifetime of PCM-based main memories through on-demand page pairing. ACM Transactions on Design Automation of Electronic Systems, 2015, 20(2): 1-23

[15] Joo Y, Niu D, Dong X, et al. Energy-and endurance-aware design of phase change memory caches//Proceedings of the Conference on Design, Automation and Test in Europe. Leuven, Belgium, 2010: 136-141

[16] Jokar M R, Arjomand M, Sarbazi-Azad H. Sequoia: A high-endurance NVM-based cache architecture. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, 24(3): 954-967

[17] Mittal S. Using cache-coloring to mitigate inter-set write variation in non-volatile caches. arXiv preprint arXiv:1310.8494, 2013

[18] Mittal S, Vetter J S. AYUSH: A technique for extending lifetime of SRAM-NVM hybrid caches. IEEE Computer Architecture Letters, 2015, 14(2): 115-118

[19] Zhou P, Zhao B, Yang J, et al. Energy reduction for STT-RAM using early write termination//Proceedings of the 2009 International Conference on Computer-Aided Design. New York, USA, 2009: 264-268

[20] Bishnoi R, Oboril F, Ebrahimi M, et al. Avoiding unnecessary write operations in STT-MRAM for low power implementation //Proceedings of the 2014 15th International Symposium on Quality Electronic Design (ISQED). Santa Clara, USA, 2014: 548-553

[21] Ahn J, Yoo S, Choi K. DASCA: Dead write prediction assisted STT-RAM cache architecture//Proceedings of the 2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA). Orlando, USA, 2014: 25-36

[22] Chen Yu-Ting, Cong J, Huang Hui, et al. Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design//Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden, Germany, 2012: 45-50

[23] Bienia C, Kumar S, Singh J P, et al. The PARSEC benchmark suite: Characterization and architectural implications//Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques. New York, USA, 2008: 72-81

[24] Binkert N, Beckmann B, Black G, et al. The GEM5 simulator. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1-7

[25] Muralimanohar N, Balasubramonian R, Jouppi N P. CACTI 6.0: A tool to model large caches. HP Laboratories, 2009: 22-31

[26] Dong X, Xu C, Xie Y, et al. NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(7): 994-1007



SHEN Fan-Fan, born in 1987, Ph.D. candidate. His research interests include computer storage system and emerging non-volatile memory.

HE Yan-Xiang, born in 1952, Ph.D., professor, Ph.D. supervisor. His research interests include trusted software, distributed parallel processing and high performance computing.

ZHANG Jun, born in 1978, Ph. D. candidate, associate

professor. His research interests include high performance computing and computer architecture.

JIANG Nan, born in 1976, Ph. D. candidate, lecturer. Her research interests include trusted software and trusted compilation.

LI Qing-An, born in 1986, Ph. D. , associate professor. His research interests include compilation optimization and embedded system.

LI Jian-Hua, born in 1985, Ph. D. , lecturer. His research interests include embedded system, emerging non-volatile memory and network on chip.

Background

Emerging non-volatile memory is a promising candidate to build large on-chip caches. It has shown the attractive features such as low leakage power, high density and better scalability. However, limited write endurance is a common problem for non-volatile memory technologies and the existing cache management policies will lead to write variation unbalance. At present, many researchers have proposed wear leveling techniques to solve this problem. These approaches can't obtain write variation dynamically and the improvement is limited.

This paper compares the existing wear leveling techniques in main memory and caches and shows the difference between them, then discusses the potential benefits of reducing the write variation. Subsequently, we propose a SEAL scheme which combines SRAM and non-volatile caches to achieve wear leveling. SEAL contains WVOM and TUBI algorithm. WVOM addresses cache inter-set write variation while TUBI focuses on cache intra-set write variation at runtime. The extensive experimental results show that the proposed approach achieves better wear leveling and notable lifetime improvement. In addition, the dynamic energy consumption is reduced and the performance is improved with an acceptable

hardware overhead.

This work is supported by the National Natural Science Foundation of China under Grant Nos. 91118003, 61170022, 61373039, 61402145, 61502346, 61640220 and 61662002, the Natural Science Foundation of Hubei Province under Grant No. 2015CFB338, the Natural Science Foundation of Anhui Province under Grant No. 1508085QF138 and the Science and Technology Project of Jiangxi Province Education Department under Grant No. GJJ150605. These projects focus on reliability, energy efficiency and performance optimization for the storage system.

Our research group has been explored many optimization approaches to improve the system reliability, energy efficiency and performance. Many papers have been published in international conference and transactions, such as Design Automation Conference, Chinese Journal of Computers, Journal of Software, Transactions on Computer, Transactions on Parallel and Distributed Systems and etc. This paper tries to solve the reliability of storage system in computer architecture and thereby improve the lifetime of computer system. Furthermore, the corresponding energy efficiency and performance are improved.