计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20150104
52(6):1225-1241,2015

新型非易失性存储器架构的缓存优化方法综述

何炎祥[1,2] 沈凡凡[1] 张 军[1,3] 江 南[1] 李清安[2] 李建华[4]
[1](武汉大学计算机学院 武汉 430072)
[2](软件工程国家重点实验室(武汉大学) 武汉 430072)
[3](东华理工大学软件学院 南昌 330013)
[4](合肥工业大学计算机与信息学院 合肥 230009)
(ffshen@whu.edu.cn)

Cache Optimization Approaches of Emerging Non-Volatile Memory Architecture: A Survey

He Yanxiang[1,2], Shen Fanfan[1], Zhang Jun[1,3], Jiang Nan[1], Li Qing'an[2], and Li Jianhua[4]
[1](*Computer School*, *Wuhan University*, *Wuhan* 430072)
[2](*State Key Laboratory of Software Engineering* (*Wuhan University*), *Wuhan* 430072)
[3](*Software College*, *East China Institute of Technology*, *Nanchang* 330013)
[4](*School of Computer and Information*, *Hefei University of Technology*, *Hefei* 230009)

Abstract With the development of semiconductor technology and CMOS scaling, the size of on-chip cache memory is gradually increasing in modern processor design. The density of traditional static RAM (SRAM) has been close to the limit. Moreover, SRAM consumes a large amount of leakage power which severely affects system performance. Therefore, how to design efficient on-chip storage architecture has become more and more challenging. To address these issues, researchers have discussed a large number of emerging non-volatile memory (NVM) technologies which have shown attractive features, such as non-volatile, low leakage power and high density. In order to explore cache optimization approaches based on emerging non-volatile memory including spin-transfer torque RAM (STT-RAM), phase change memory (PCM), resistive RAM (RRAM) and domain-wall memory (DWM), this paper surveys the property of non-volatile memory compared with traditional memory devices. Then, the advantages, disadvantages and feasibility of architecting caches are discussed. To highlight their differences and similarities, a detailed analysis is then conducted to classify and summarize the cache optimization approaches and policies. These key technologies are trying to solve the high write power, limited write endurance and long write latency of emerging non-volatile memory. Finally, the potential research prospect of emerging non-volatile memory in future storage architecture is discussed.

Key words non-volatile memory (NVM); storage technology; computer architecture; cache; optimization approach

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,传统存储器件的漏电功耗问题日益严峻,如何设计高能效的片上存储架构已成为重要挑战.为解决这些问题,国内外研究者讨论了大量的新型非易失性存储技术,它们具有非易失性、低功耗和高存储密度等优良特性.为探索 spin-transfer torque RAM(STT-RAM),phase change memory(PCM),resistive RAM(RRAM)和 domain-wall

收稿日期:2015-02-05;**修回日期**:2015-04-20
基金项目:国家自然科学基金重点项目(91118003);国家自然科学基金面上项目(61170022,61373039);国家自然科学青年基金项目(61402145);高等学校博士学科点专项科研基金项目(20130141110025l2);安徽省自然科学青年基金项目(1508085QF138)
通信作者:何炎祥(yxhe@whu.edu.cn)

memory(DWM)四种新型非易失性存储器(non-volatile memory, NVM)架构缓存的方法,对比了其与传统存储器件的物理特性,讨论了其架构缓存的优缺点和适用性,重点分类并总结了其架构缓存的优化方法和策略,分析了其中针对新型非易失性存储器写功耗高、写寿命有限和写延迟长等缺点所作出的关键优化技术,最后探讨了新型非易失性存储器件在未来缓存优化中可能的研究方向.

关键词 非易失性存储器;存储技术;计算机体系结构;缓存;优化方法

中图法分类号 TP303; TP333

近年来,随着硬件技术的飞速发展,功耗问题导致处理器主频不能进一步增加,转而走向多核处理器设计.多核技术日趋成熟及多线程技术的广泛应用,处理器性能得到极大提高;但是,主存的访问速度却较为缓慢,严重阻碍了计算机系统整体性能的提升.处理器和主存之间的性能差距也逐渐增大,"存储墙"的问题日益严峻.缓存在一定程度上能够缓解访问速度不匹配的问题.图 1 显示了缓存的访问速度、代价和容量与缓存层次结构之间的关系.

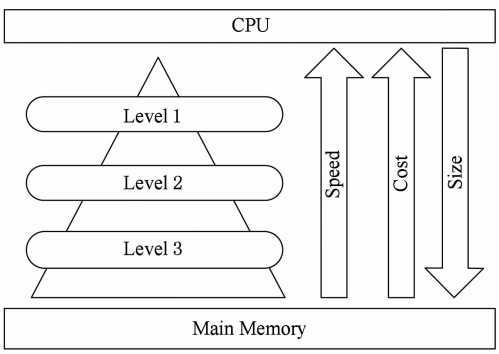


Fig. 1 Cache hierarchy.

图 1 缓存层次结构

随着半导体工艺和处理器技术的发展,片上集成越来越多的核是发展趋势,为了满足核心数增加带来的数据访问压力,进而又需要容量更大的缓存,例如 Intel Haswell-EP Xeon E5-2699 v3 系列处理器的最后一级缓存(last-level cache, LLC)大小为 45 MB[1].以前通常采用 SRAM 存储器件架构片上缓存,因为它具有访问速度快、动态功耗低及耐久性(endurance)高等优点.然而,随着半导体特征尺寸的进一步降低,基于传统 CMOS 工艺的 SRAM 的漏电功耗急剧增加并逐渐占据主导地位.对于大容量的缓存,它将耗费大量的芯片面积及功耗,例如 StrongARM 处理器的片上缓存将消耗总功耗的 30%[2].基于 SRAM 设计的片上缓存已经无法满足现代计算机系统对高性能和低功耗的要求,因此,探索新型存储技术来提升缓存的性能及降低功耗具有十分重要的意义.

新型非易失性存储器(non-volatile memory, NVM)的出现得到了学术界和工业界的广泛关注,并为计算机存储技术提供了新的解决方案.新型 NVM 是很有希望取代传统存储器件(如 SRAM 和 DRAM)的,因为它具有集成度高、漏电功耗低、访问速度快、非易失性等优点.由于新型存储器件的制造工艺和设计原理不同,它们的大小、读写速度、读写功耗和存储寿命都略有不同.目前研究比较活跃的 NVM 有铁电存储器(ferroelectric RAM, FeRAM)、磁性存储器(magnetic RAM, MRAM)、自旋转移力矩存储器(spin-transfer torque RAM, STT-RAM)、相变存储器(phase change memory, PCM)、阻变存储器(resistive RAM, RRAM)、赛道存储器(domain-wall memory 或 racetrack memory, DWM)等[3-4].随着研究工作的深入,这些新型存储技术已逐渐从原型设计阶段走向产品产业化阶段.例如 Samsung 公司推出 65 nm 工艺的 512 MB 容量的 PCM 芯片已经在手机存储卡中使用;Micron 公司研制的 45 nm 工艺的 1 GB 容量的 PCM 芯片用于第二代低功耗内存(LPDDR2)中;Everspin Technologies 公司研制了第一款 64 MB 的 STT-RAM 商业产品用于第三代低功耗内存(LPDDR3)中.然而新型 NVM 完全取代传统存储器件仍有许多问题亟需解决:1)新型 NVM 特有的属性问题.如写的寿命有限、写的功耗大和读写速度不一样等,这些都阻碍新型 NVM 在计算机系统中的广泛使用.2)新型 NVM 的管理方法.由于新型存储器件自身的属性问题,传统的存储系统管理方法已不适用,需要针对新型 NVM 的特点优化相应的算法,以提高计算机系统性能和降低功耗.3)新型 NVM 的引入会带来一些新的问题.一些意想不到的问题会在研究过程中不断涌现[3],因此,运用新型非易失性存储器架构的计算机存储子系统是一个非常有挑战和值得研究的问题.

本文主要从新型 NVM 架构技术的角度探讨如何解决上述问题,重点讨论研究非常热门的新型存储技术:STT-RAM, PCM, RRAM 和 DWM.讨论的范围包括:1)仅讨论新型存储技术在缓存中应用

的优化方法,尽管这些存储技术可以用于主存、磁盘等存储子系统中,但和缓存优化方法的角度不同;2)仅讨论体系结构中的创新优化方法,重点关注优化方法而不讨论实验平台、测试集和电路级设计;3)重点讨论当前主流的研究方法.本文的主要内容如下:

1)对比了新型 NVM 的基本属性、优缺点及适用性;

2)从多个角度对缓存优化方法进行了分类,并针对 4 种新型 NVM 中使用的缓存优化方法进行了分析;

3)对新型 NVM 在未来缓存架构中的优化方法进行了探讨.

1 新型 NVM 简介

为了深入了解新型 NVM 架构的特性,图 2 对比了不同存储技术的读写延迟和功耗等关键指标.表 1 从工艺尺寸、容量、读写速度、寿命、保留时间、优缺点及适用性等多个方面进一步展示了各存储技术的特点.

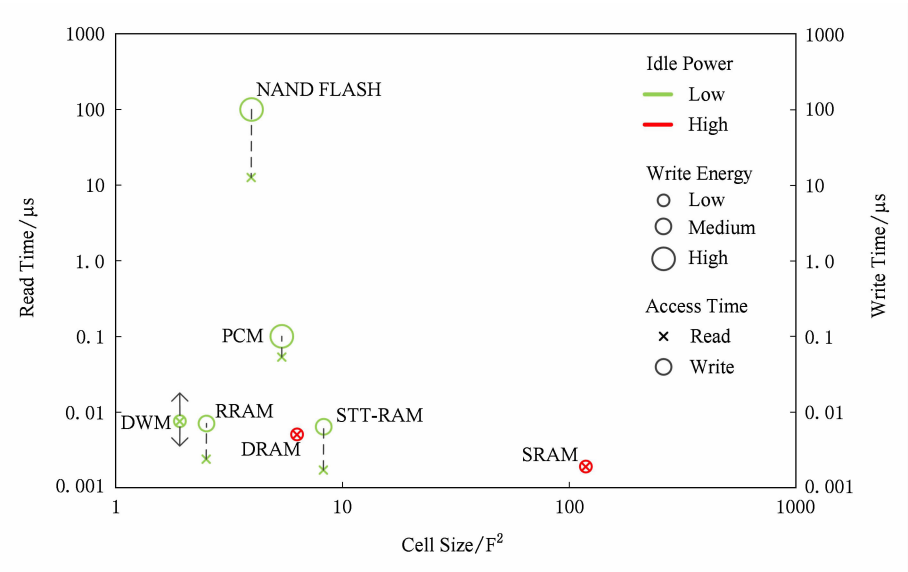


Fig. 2 Comparison of different memory technologies.

图 2 不同存储技术对比

Table 1 Characteristics Comparison of Different Memory Technologies[5-11]

表 1 不同存储器件的特性对比[5-11]

Criterion	SRAM[5]	STT-RAM[5]	PCM[6-7]	RRAM[8]	DWM[9-11]
Maturity	Product	Product	Product	Prototype	Research
Technology Node/nm	32	32	5	11	15
Cell Size/F^2	120-200	6-50	4-12	4-10	1-4
Capacity/MB	≤32	≤64	$\leqslant 8\times10^3$	$\leqslant 10^6$	$\leqslant 10^6$
Speed(Read/Write)	Very Fast	Fast/Slow	Slow/Very Slow	Fast/Slow	Fast/Slow
Endurance	10^{16}	10^{15}	10^9	10^8	10^{16}
Non-Volatile	No	Yes	Yes	Yes	Yes
Leakage Power	High	Low	Low	Low	Low
Dynamic Energy (Read/Write)	Low	Low/High	Medium/High	Low/High	Low/High
Retention Period	Long	Long(unless relaxed)	Long	Long	Long
Advantages	Fast	Non-Volatile, Low Leakage Power Potential to Scale	Non-Volatile, Low Leakage Power, High Density	Non-Volatile, Low Leakage Power, High Density	Non-Volatile, Low Leakage Power, High Density
Disadvantages	Low Density, High Leakage Power	Long Write Latency, High Write Energy, Poor Stability	Low Speed, Low Endurance	Low Endurance, High Write Energy, Poor Stability	High Write Energy
Applicability	Cache	Cache	Main Memory	Main Memory	Flash Memory

从表 1 可以看出,和 SRAM 相比,NVM 具有存储密度大、非易失性、静态功耗低和可扩展性强等优点;当然也有不足,写延迟和写功耗均比 SRAM 高,这给新型 NVM 的大规模应用提出了挑战.

1.1 新型 NVM 的原理

1) STT-RAM[12-13]. 使用磁性隧道结(magnetic tunnel junction, MTJ)存储数据位,MTJ 包含 2 个磁性层:参考层(reference layer)和自由层(free layer),参考层的方向固定,通过自由层的方向变换来决定数据位的值,当 2 层平行时存储数据位为 0,当两层反向时存储数据位为 1. 由于 MTJ 材料自身属性的原因,一次写操作需要花费大量的时间,因此写延迟和写功耗偏大. 和 PCM,RRAM 相比,它拥有更长的寿命,几乎接近 SRAM.

2) PCM[6]. 由硫族化合物材料构成的,该材料根据温度的变化可以形成结晶状态和非结晶状态,这样可以用来存储数据,结晶状态时存储数据 1,非结晶状态时存储数据 0. 它的写 1 操作是一个结晶的过程,需要一个较长时间的电脉冲进行加热,当温度达到结晶温度以上和熔化温度以下时,这样就形成了结晶;相反,写 0 的过程是使用作用时间短和强度高的电脉冲,使该材料迅速上升到熔点并立刻释放热量熔化,这样就形成了非结晶状态. 可见,它的写 1 和写 0 的过程是不对称的. PCM 具有工艺尺寸小、非易失性、功耗低和存储密度高等特点.

3) RRAM[14]. 通常是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. 其中,电阻转变存储技术采用单极性电阻转变和双极性电阻转变 2 种模式,当给阻变存储器施加高电压时,电阻层发生电阻转换,从而形成高阻态和低阻态,其中高阻态存储数据 0,低阻态存储数据 1. RRAM 具有存储密度高、漏电功耗低、非破坏性存取和读速度快等特点.

4) DWM[9]. 通过控制畴壁(domain wall)在磁性纳米线上的移动来实现. 一段旋转一致的电流沿着纳米线移动磁畴(magnetic domain),当磁畴经过纳米线时,就可以存储或修改数据位. 每个磁畴可以存储一位数据,一个存储单元包含多个磁畴,因此一个存储单元即可存储多个数据位. DWM 具有存储密度高、超低功耗和价格低廉等特点,优于目前市面上的闪存和固态硬盘.

1.2 新型 NVM 的优缺点

1) 新型 NVM 的优点

① 新型 NVM 相比传统存储器件具有非易失性的特点,即使系统掉电,NVM 保存的数据也不会丢失并且 NVM 可以长期保存数据.

② 在漏电功耗方面,新型 NVM 具有极低的漏电功耗,特别适合于大容量的存储器,可以大幅度减少存储系统的功耗.

③ 在存储密度方面,新型 NVM 拥有更高的存储密度,与相同芯片面积的传统存储器比,新型 NVM 可存储更大容量的数据,如 RRAM 和 DWM 的存储容量可以达到 TB 级别.

④ 新型 NVM 的可扩展性强,可以根据需求定制存储器的架构,如 STT-RAM 在设计的时候可以通过放松其非易失性来优化其功耗和写延迟. 由此可见,新型 NVM 的灵活性更高.

2) 新型 NVM 的缺点

① 新型 NVM 的读写操作是不对称的,写延迟比读延迟长,这造成了读写不均衡的问题同时写延迟高于传统存储器.

② 在动态功耗方面,新型 NVM 的写操作功耗非常高,这是由新型 NVM 的存储材料和原理所决定的.

③ 从耐受性方面考虑,PCM 和 RRAM 具有写耐受性差的缺点,写的次数有限,影响了存储器件的使用寿命. STT-RAM 和 DWM 的写耐受性接近传统存储器件.

1.3 新型 NVM 的适用性分析

新型 NVM 在存储子系统各个层次中都有应用,如在缓存、主存和外存等存储架构中使用. 然而从新型 NVM 各自的特点和存储层次结构的不同访问需求考虑,STT-RAM 最适合架构缓存,因为缓存是 CPU 访问最频繁的,对读写速度和次数都有要求,STT-RAM 的写耐受性高,故适合做缓存;PCM 虽然也有用于架构缓存和外存方面的研究,但是它最适合于设计主存,因为主存可以容忍 PCM 的高访问延迟和写耐受性,然而它的高存储密度可以弥补它的不足;RRAM 有应用于缓存架构的,它和 PCM 一样存在写次数有限,但比 PCM 容量大、访问速度快,也适合做主存;DWM 存储技术尚未成熟,还属于研究阶段,有研究者将其应用于缓存架构,由于它具有显著的优点,存储容量非常大,写的次数接近 DRAM,故它是替代闪存和固态硬盘最理想的存储技术.

2 基于新型 NVM 的缓存优化方法

根据第 1 节的分析可知,新型 NVM 在漏电功耗、存储密度和可扩展性等方面优于传统存储器件.

新型 NVM 的存储特性和参数指标均类似于传统存储器,因此研究者试图根据新型 NVM 的访问特性架构存储子系统,通过模拟和仿真的方式实现相应的优化策略.主要从多种角度解决新型 NVM 的写功耗高、写次数有限和写延迟大等问题.本节重点讨论新型 NVM 架构的缓存优化方法,首先对所有的缓存优化方法从整体上进行分类和总结,接着从细粒度分析 4 类新型 NVM 中各自的缓存优化方法.

2.1 缓存优化方法分类与总结

通过对新型 NVM 的缓存优化方法探索,从整体上看,表 2 分类和总结了当前主流的研究方法,主要对 4 类新型存储技术在缓存中的应用、在缓存中使用的层次结构(一级、二级或最后一级缓存)、混合缓存架构方法、缓存优化的目标(节省能耗、提高性能、延长使用寿命和增强可靠性)及在 CPU 和 GPU 中的使用等多个角度进行了总结.从表 2 文献引用可以看出,研究者重点集中在研究使用 STT-RAM 或混合的方式架构缓存,主要在 CPU 及其中间或最后一级缓存中使用,新型 NVM 不适合架构读写频繁的一级缓存,缓存优化方法追求的主要目标是降低功耗和提升性能;其他的研究者在探索另外 3 种新型存储技术在缓存中的应用,并给出了自己的见解和优化方法;近年来,研究者开始关注新型存储技术在 GPU 缓存中的应用,使 GPU 的能耗大幅度下降,性能显著提高.这些分类和优化方法给以后的研究者提供了一种思路,具有借鉴和参考意义.

Table 2 Classification and Summary of Cache Optimization Approaches

表 2 缓存优化方法分类与总结

Criterion	Approaches	References
NVM Technologies	STT-RAM	Ref. [3-8,10,12-69]
	PCM	Ref. [3,6,10,17,33-34,70-73]
	RRAM	Ref. [10,14,33-34,61]
	DWM	Ref. [9-11,74-80]
Cache Levels	First-Level Cache	Ref. [15,35,38,49,58,71,75-76,78]
	Middle or Last-Level Cache	Ref. [1-80]
Hybrid Architectures	SRAM+STT-RAM	Ref. [6-8,18,32,40-41,45,49-51,54,56,60]
	DRAM+STT-RAM	Ref. [24,61,64]
	PCM+STT-RAM	Ref. [17]
	SRAM+PCM	Ref. [6,71,73]
	SRAM+DWM	Ref. [9]
Cache Optimization Goals	Saving Energy	Ref. [5-9,12-14,16,18-19,21-32,35-38,40-41,43,45-47,49-63,66-68,70-80]
	Improving Performance	Ref. [6,8,21,25,27,29,38-40,42,46-47,53,56,62-69,73-74,77-80]
	Improving Lifetime	Ref. [17,26,32-33,39,45,48,51,72]
	Reliability	Ref. [42,44,69]
Applications	CPU Caches	Ref. [1-80]
	GPU Caches	Ref. [40,46,61-63,70,79-80]

为了进一步探索缓存优化的通用性方法和研究角度,对于新型存储技术存在的问题,从细粒度的技术层面看,主流的优化方法可以划分为如下 3 种:

1) 通过减少 NVM 的写操作来降低功耗

由于 NVM 的写功耗大,研究者主要从位级、访问级、混合缓存等方面减少对 NVM 的写操作.在位级,通过在标志数组(tag array)中增加 all-zero-data 标志来识别所有位都是零的数据,当检测到此标志时避免执行写操作[30];同时也可以通过位级感知写操作来避免不必要的写操作,如许多将要写的位已经存储在缓存中,那么没必要再次写入该位,可以通过先读后写的方式判断该位是否一样,从而减少写的次数和功耗[13];第 1 次写操作序列是必要的,第 2 次再写同样的数据就是不必要的[23];当识别出写操作时,可以以低电流的方式执行写[36];对于缓存高数据位经常不变,低数据位变化频繁的,可以减少低数据位相同的写操作[37].在访问级,将脏数据(dirty data)尽量保留在 L1 中,将脏数据保留多个时钟周期

后再替换到 L2 或最后一级缓存[19];通过预测的方式将死写(dead write)操作绕过最后一级缓存,最大程度避免这些块写入缓存[59];通过编译指导数据分配,将大量的写操作交由快速写存储处理[29]. 在混合缓存方面,研究者充分利用混合缓存各自的优点减少对 NVM 的写操作,如在 SRAM 中执行写操作、在 STT-RAM 中执行读操作,可以通过编译器指导的方式实现[8,45,55];同时也可以动态地将写密集型数据由 STT-RAM 迁移到 SRAM 中并减小迁移开销[32,55,60];通过缓存一致性协议的状态位来动态迁移缓存块[58]. 这些方法都在最大程度地减少对 NVM 的写操作,从而达到减少功耗的目的.

2) 通过耗损均衡(wear-leveling)来延长 NVM 的寿命

减少对 NVM 的写操作在一定程度上能延长 NVM 的寿命. 由于 NVM 的写操作次数有限、缓存中的写操作不均匀,这造成某些缓存单元会优先损坏,这样就缩短了 NVM 的寿命. 为了解决这个问题,研究者提出了耗损均衡的方法,将写操作均匀地分布到缓存块中,减少缓存组间和组内(inter and intra-set)的写波动(write variation)来延长 NVM 的寿命[33-34,48],也有通过缓存分区和访问感知策略来减少缓存不均衡的耗损[51].

3) 通过减少 NVM 的写延迟来提高性能

由于 NVM 的物理特性导致了其写延迟高,研究者试图从体系结构、数据预取及 NVM 访问的随机性等角度减少写延迟. 如在缓存的微架构中加入写缓冲器(write buffer),这样可以隐藏缓存访问的长延时[65];通过减少数据预取间的冲突问题也可以减少访问延迟[28,67];通过概率性访问的方式动态地调整写脉冲的周期来减少写延迟[42]. 这些方法都能有效地提高缓存的性能.

2.2 基于 STT-RAM 的缓存优化方法

2.2.1 概 述

STT-RAM 具有显著的特性,为其在缓存领域的应用奠定了深厚的基础;但它也存在一些缺陷,特别是写功耗高、写寿命有限和写延迟大,这些阻碍了它广泛应用于现有的计算机系统,却促进了研究者提出各种方案和策略来弥补它的不足.

学术界已经从多个角度展开了研究,表 3 对 STT-RAM 的优化标准进行了详细分类,并给出了各类别中采用的优化方法. 优化标准包括减少写操作次数来降低功耗、减少写延迟来提升性能、通过耗损均衡来延长寿命、通过混合缓存来弥补写操作的不足、释放 STT-RAM 的易失性来提升性能、解决读写不对称问题、使用多层单元(multi-level cell)提高存储密度和在 GPU 中的应用来降低功耗. 缓存优化标准涉及的方法包括在位级感知并避免不必要的写操作[13,23,36-37]、通过数据预取技术减少写操作冲突[28,67]、通过数据分配方法均衡缓存组内组间的访问[33-34,48]、利用编译指导数据分配方法[8,15,29,45,52,55]、修改存储体系结构[6-7,18-19,25,30,40,49,51,62-63,65]、预测缓存死写操作[41,50,59]、通过计数器控制缓存刷新次数[38,52-53,57]、数据迁移感知方法[32,55,60]等. 这些方法极大地推动了 STT-RAM 的发展,给研究人员和技术人员提供了科学的理论参考.

Table 3 Classification of Cache Optimization Approaches Based on STT-RAM

表 3 基于 STT-RAM 的缓存优化方法分类

Criterion	Approaches and References
Minimizing Write Operation	Early Write Termination[13] Reduce Dead Write[59] Compiler[15,29] Architecture[19,30] Write Current in Bit-Cell[23] Low-Current Probabilistic Write[36] Lower-Bits Cache[37]
Reducing High Write Latency	Two Orthogonal Techniques[28,67] Probabilistic Design[42] Architecture[65] Obstruction Aware[66]
Improving Cache Lifetime	By Coding Bit Cell[26] Reduce Inter and Intra-Set Write[33-34,48] Partition-Level[51]
Hybrid Caches	Architecture[6-7,18,49,51] Compiler[8,45,55] Data Migration[32,55,60] Prediction Based Methods[41,50] Cache Coherence[58]
Volatile STT-RAM	Relax Non-Volatility[12] Architecture[25] A Counter Based Dynamic Refresh[38] Compiler[52] Cache Coherence[53,57]
Solving Read/ Write Asymmetry	Reduce Write Latency[20] Address Process Variation[21-22] Log Style Write[68] Dynamic Configuration[69]
Using Multi-Level Cell	Line Pairing and Line Swapping[27] Set Remapping Scheme[39] Unleashing the Potential of MLC[47]
Apply to GPU	Architecture[40,62-63] Using for Shared Memory[46]

2.2.2 减少 STT-RAM 的写操作次数

减少对 STT-RAM 的写操作能有效降低能耗,其优化方法和策略众多,本节仅讨论主要优化技术中的创新思想,为研究人员提供一种思路.

文献[13]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,许多相同的数据位已经在缓存中,这些写操作是多余的,应该避免这类写操作.该方法设计了一个 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,读操作通过电流感知 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.这种方法能有效地减少写能耗,但是引入了硬件开销和额外的读操作开销.

文献[19]提出了一种基于 STT-RAM 架构的缓存优化方法来减少动态功耗.该方法的思想是将脏行(dirty lines)数据保留在 L1 中,这样可以避免过早地将频繁访问的数据写入 L2 和 LLC 中;同时提出了写偏向(write biasing)和写缓存(write cache)2 种优化技术来减少写操作.write biasing 技术是通过缓存替换算法将 dirty lines 数据尽可能长地保留在 L1/L2 中;而 write cache 技术是在 L1 和 L2 之间加入一个小的缓存,专门用于存储从 L1 中替换出来的 dirty lines 数据,这样可以缓解从 L1 中写入数据到 L2 中的次数.这种架构方法减少了动态功耗,但是增加了小缓存块的硬件开销以及缓存管理策略的开销.

文献[23]提出了一种位级感知写操作的行为来减少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,这个过程可以由位单元的值检测(bit-cell value detection)和实际的写操作(actual write operation)2 部分组成.第 1 部分发送一段电流读取 MTJ 的状态位,这样便可知道当前存储的位单元的值;第 2 部分则通过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.该方法也采用了先读后写的方式避免不必要的写功耗,但是增加了硬件开销,需要设计和修改相应的位电路.

文献[30]提出了一种探索许多应用程序处理大量零数据(所有的数据位都存储的 0)的架构技术.该技术在缓存 tag array 中增加了 all-zero-data 标志位,带有此标志的数据值为零.当执行写操作时,缓存先检查待写的数据是否都是零,如果检测到,则设置 all-zero-data 标志,仅将非零的数据写入 STT-RAM 中;当执行读操作时,仅读取缓存中非零标志的数据.通过这种方式能减少大量的写操作,降低功耗,但是每个数据行都会增加一个标志位,增加了硬件开销,同时 all-zero-data 检测也会有一定的代价.

文献[36]提出了一种低电流的概率性写操作方法,即通过调低写电流脉冲来降低功耗.然而电流过低会导致写操作失败,为解决该问题,该文献给出了一个迭代式框架:第 1 步先以低电流执行写操作;第 2 步读取刚写入的数据;第 3 步与初始写入的数据对比来决定本次写操作是否成功,如果对比发现不成功则再从第 1 步开始执行.所有的写入操作都按照这个迭代式框架执行,准确无误便写入该数据;同时为了进一步保证写过程的正确率,该方法引入了伯努利概率方程,精确地预测每次写操作.上述方案通过动态调整写电流脉冲的强弱来降低写功耗,比较新颖,但迭代式框架的频繁执行会降低系统的性能.

文献[59]提出了一种死写预测(dead write prediction)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,如果能减少这些死写操作可以大幅度地降低缓存功耗.为了达到这个目的,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.因为抽样预测的精度非常高,该方法能减少大部分最后一级缓存的死写操作,因此降低了写功耗,然而不可避免地将死写操作转移到了 L1 中,而 L1 的缓存空间和资源是非常有限的,故带来一定的额外开销.

2.2.3 降低 STT-RAM 的写延迟

STT-RAM 的高写延迟会严重影响缓存的性能,下面探讨主要的优化写延迟的方法.

文献[28,67]提出了通过提高数据预取效率来管理和优化 STT-RAM 缓存的方法.文献[28]给出了基本优先分配和优先启动(priority boosting)的方法,基本优先分配方法是根据不同类型的访问请求(读、写、预取和写回等)的重要性来分配优先级,重要的请求先执行,这样可以减少各个请求之间的访问冲突,降低等待延迟;priority boosting 方法是用于区分应用程序访问请求的多少,对于访问请求少的应用程序分配较高优先级,这样可以加速非密集

型程序请求的速度,提升整体系统的性能.文献[67]提出了请求优先和混合本地全局预取控制的方法.由于数据预取机制,写请求的次数迅速增长,这样会增加其他请求的等待时间,使整体访问周期延长,请求优先方法认为读请求具有最高优先级应马上处理、写请求次之,预取请求和写回请求的优先级低些,因为预取和写回的数据并不会立刻使用.混合本地全局预取控制方法包含本地预取控制和全局预取控制,本地预取控制是针对每个核的预取策略;而全局预取控制是针对所有核和共享缓存的预取策略,它周期性地抽取每个核的预取频率和 LLC 的全局访问频率来评估预取精度,该评估结果将指导下一次预取操作.通过提高数据预取的效率能有效减少访问请求的冲突,降低整体等待周期,提高缓存的性能,该方法依赖于请求分类的准确性和程序的访问特点,对于写密集型程序的效果不佳.

文献[42]提出 2 种概率性设计来减少写延迟和写失败.2 种方法分别为自适应周期性的写后再校验(write-verify-rewrite with adaptive period, WRAP)和写操作时仅检验 1(verify-one-while writing, VOW).WRAP 是一种类似先读后写缓存校验方式,如果写失败该校验方式会重复执行,这样就会存在一个写脉冲周期,为了权衡写脉冲周期和多次重复执行次数间的关系,WRAP 给出了 τ_{opt} 参数动态地调整写脉冲周期,这样就可以减少额外的写脉冲周期.VOW 是为了解决 0 和 1 转换电压感知复杂和读出放大器需要预充电的问题;同时根据统计学的观点写 0 要比写 1 快,在相同的写脉冲周期中,如果所有写 1 的操作都成功了,那么写 0 出错的概率极低,VOW 仅校验写 1 的操作,这样减少了校验周期.该方案能有效地减少写延迟和提高写操作的稳定性,但也会引起额外的硬件开销和动态调整策略开销.

文献[65]从体系结构级探讨了 STT-RAM 替换 SRAM 或 DRAM 对芯片面积、缓存性能和功耗的影响.STT-RAM 相比 SRAM 和 DRAM 拥有更低的漏电功耗和更大的容量,但写延迟较大.在相同芯片面积下,STT-RAM 的写延迟可以通过写缓冲来隐藏,数据在写入 STT-RAM 前先写入该缓冲,提高了缓存的性能,然而对于写密集型程序,该写缓冲无法满足需求,对性能提升不大.

2.2.4 STT-RAM 的耗损均衡

STT-RAM 的写次数有限,下面探讨通过耗损均衡的方法提高缓存的使用寿命.

文献[33-34,48]提出了一个新的缓存管理策略来减少组间和组内的写波动,该策略有 swap-shift 和 probabilistic set line flush(flush 指将数据写入缓冲区而不写入缓存行中)2 个特点,分别用来减少组间写波动和组内写波动.swap-shift 方法将每 2 个缓存组映射成一对,每次在对应的组间进行交换,同时如果某个组写的次数过多,那么它就和相邻的下一个组交换.这个方法可以使组间写均衡化,但是无法解决组内写不均衡问题,因为按照缓存管理策略,访问频繁的数据将一直写入行首.probabilistic set line flush 方法是以一定的概率 flush 频繁访问的数据,通过一个全局计数器记录频繁访问的数据,该数据访问越频繁那么被 flush 的概率就越大,这样就减轻了频繁访问的数据一直访问某些位而造成位损坏.上述策略从整体上可以均衡写操作,但是对于写密集型程序需要足够的缓冲区才能顺利地保存频繁访问的数据.

文献[51]首先提出了 SRAM, STT-RAM 和 SRAM/STT-RAM 的混合架构方法,接着提出了分区级(partition-level)耗损均衡方法和访问感知策略来减少不同分区间的不均衡耗损.访问感知策略主要针对 STT-RAM 的写管理和 SRAM 的读管理.STT-RAM 的写管理用于减少写入的次数和使写操作均匀分布,其管理过程是如果写请求到来并且 SRAM 中存在可写的行,那么直接将写请求重定向到 SRAM 中,如果 SRAM 中没有可写的行则将写请求重定向到其他 STT-RAM 行中.SRAM 的读管理过程和 STT-RAM 的写管理过程比较类似,将 SRAM 中的数据拷贝到 STT-RAM 中,然后在 STT-RAM 中执行读操作.partition-level 耗损均衡方法将混合缓存分为 4 大块,每大块包含 1 个 SRAM 块、2 个 STT-RAM 块和 1 个 SRAM/STT-RAM 块,根据 SRAM 和 STT-RAM 访问的次数动态地调整分区的路数.上述架构方法和管理策略能够提高缓存的寿命和降低系统功耗,但是需要花费一定的管理代价.

2.2.5 混合缓存架构

混合缓存架构主要是 SRAM 和 STT-RAM 组合在一起利用 SRAM 的写速度快和 STT-RAM 的读取速度快、存储密度高和漏电功耗低等优点相互弥补各自的不足,下面探讨主要的混合缓存的架构和优化方法.

文献[7]提出了一种动态可重构的混合 SRAM/STT-RAM 架构方法.该架构将缓存行的数据数组(data array)分配给 SRAM 和 STT-RAM,同时增加了功耗门控装置,它可根据命中计数器的上限动

态地开关 SRAM 和 STT-RAM 的路数,从而节省功耗.

文献[45]提出了静态方法和动态方法结合起来优化块的放置方法.该方法使用编译器提供静态标识指导数据的初始放置;然后根据运行时缓存的访问行为来修正该标识;最后将写密集的数据放到 SRAM 中,写不密集的数据放到 STT-RAM 中.该方法能降低混合缓存的功耗,但是依赖于编译器和硬件的支持.

文献[50]提出了一种写强度预测的方法来减少混合缓存中 STT-RAM 的写操作.预测器追踪那些倾向于写密集的指令,接着利用该信息来预测缓存块的写强度,写强度高的将载入到 SRAM 中,其他的将载入到 STT-RAM 中.该方法能减少写功耗,预测器的准确性将影响写操作的放置.

文献[58]提出了一种基于 MESI 缓存一致性协议的缓存块动态分配策略.该文献首先给出了一级缓存的混合架构方法,接着根据缓存块的状态在 SRAM 和 STT-RAM 之间动态地迁移缓存块,如缓存块是 M(modified)状态时应该放置到 SRAM 中,而 S(shared)和 E(exclusive)状态时应该放置在 STT-RAM 中.该策略能减少缓存的功耗,但是由于一级缓存读写访问频繁,缓存状态变化快,会带来一定的迁移开销.

文献[60]提出了一种混合缓存自适应放置和迁移块的策略.LLC 的写访问可以分为 3 类:预取写(prefetch-write)、核写(core-write)和需求写(demand-write),然后根据访问模式预测器指导块的放置和迁移操作.通过统计实验结果数据分析访问模式可知,对于预取写块,数据写入缓存后零次或少量的读应该放置在 SRAM 中,其余的放在 STT-RAM 中;对于核写块,数据写入缓存后立刻读的适合放在 SRAM 中,其他的适合放置在初始位置;对于需求写的块,数据写入缓存后没有读的没必要写入 LLC,其他的适合放置在 STT-RAM 中.为了动态地调整缓存的写强度,该方法在程序执行过程中可以动态感知写强度并迁移块.该策略是根据 CPU 测试集的统计结果指导缓存块放置的,试用范围有限且精度有待提高.

2.2.6 易失性 STT-RAM 缓存架构

为了权衡 STT-RAM 的数据保留时间和写延迟之间的关系,文献[12,25]提出了释放 STT-RAM 的非易失性并寻找最优的数据保留时间来架构易失性 STT-RAM 缓存,减少写延迟,提高性能;然而数据保留时间过低会导致数据丢失,那么需要一定的刷新机制保证数据的正确性,下面探讨主要的易失性 STT-RAM 缓存架构的优化方法.

文献[38]提出了一种基于 STT-RAM 的 L1 和低级缓存(lower level cache)设计,各级缓存拥有不同的数据保留时间.由于 L1 读写访问频繁、对读写效率要求高,故 L1 缓存使用低的数据保留时间,同时通过计数器控制缓存动态刷新以防止数据丢失;对于低级缓存容量相对较大,访问速度可容忍,故设计的缓存数据保留时间较高,并且缓存组中不同缓存路数(cache ways)也含有不同的数据保留时间,为了保证性能和功耗,数据迁移方法将数据在不同保留时间的 cache ways 上移动.上述方案可以充分利用 STT-RAM 的不同数据保留时间,使性能和能效达到最大化.

文献[52]提出了一种编译器辅助的易失性 STT-RAM 刷新最小化策略.该策略在编译时重新分配数据并指出了主动刷新存在的问题;接着提出了线性规划和启发式算法的解决方案来使主动刷新最小化,为了进一步减少缓存刷新的次数,该策略又提出了一种解决 2 次连续访问距离非常长的 N 次刷新方法.上述几种方案综合起来可以减少编译时和运行时的缓存刷新次数,降低功耗,但是线性规划的实现存在一定的时间复杂性.

文献[57]提出了一种基于缓存一致性协议的自适应刷新策略来减少易失性 STT-RAM 刷新操作的次数.该文献首先分析了缓存刷新操作对系统性能的影响;接着提出了不刷新策略和给定刷新次数的策略.不刷新策略指每次检查缓存块是否有效,若无效则不用刷新;而给定刷新次数的策略是指每个缓存块刷新 N 次.经过上面的刷新策略,有些缓存块会丢失数据,为了维护缓存的一致性,该文献在 MESI 缓存一致性协议中增加了 W 状态来维护这个过程,从而保证了整个缓存中数据的一致性,同时也最大化地减少了刷新操作.上述策略能降低多核系统的功耗同时提高性能,但是维护缓存的一致性会有一定的开销.

2.2.7 STT-RAM 的读写不对称性

读写不对称性是指 STT-RAM 中写 0 和写 1 的时间不同,那么 1 到 0 和 0 到 1 状态转换所需要的时间也不相同且写的速度偏慢.为解决这些问题,下面探讨主要解决读写不对称的缓存优化方法.

文献[20]提出了一种使用冗余块的读写不对称架构方法.该方法是在每个缓存行引入额外的位,当执行写操作,首先检查缓存行是否有预设为 0 的位,

如果存在,那么当前写操作并不直接写入0数据,而是将预设位标记为实际的数据位,然后再将其移动到数据位实际的位置;如果不存在,那么直接将数据位写入其实际位置.例如当缓存行预先存储着0状态,当某个缓存行发生写操作时,可以不需要1到0的转换过程,这样写的速度大幅度提高.该方案通过增加冗余位来减少STT-RAM写慢的问题,减少1到0的转换时间,隐藏了部分写延迟,但增加了一定的缓存位开销.

文献[22]提出了一种异步读写不对称终止的方法,根据读写不对称的行为来终止写操作.当存储单元处于稳定的1状态时,写电路可以异步地关闭.基于这样的观察结论,该文献给出了时钟信号控制方法和自动同步(self-timing)的方法,时钟信号控制方法发出一个时钟信号驱动写终止,接着self-timing方法根据MTJ存储单元的信息产生一个写终止信号,这样可以减少不必要的写电流,降低写功耗,当然也会产生一些控制电路的开销.

文献[68]提出了一种通用的log style write方法来减少STT-RAM的写0和写1状态转换不对称性问题.该方法将写0和写1分2个阶段分别处理,初始时只有写1操作,当新数据要写入缓存时只执行写0操作.例如一个8路相连的缓存组,在每组中增加一路并记为log way,它初始时被程序化为1,每当写入新数据时待写入的那一路标记为无效,而将新数据中的0位写入log way中,在下一轮写操作前无效的那一路重置为1并记为log way,每次访问都循环重复这个过程.这样每次新数据的写操作中仅有写0操作,那么写延迟和写功耗都大幅度减少,然而却带来了12.5%的存储空间开销和log way重置为1的开销.

2.2.8 多层存储单元的STT-RAM

多层存储单元STT-RAM是指每个存储单元可以存储多个数据位,存储密度提高了,这对于构建大容量低功耗的片上缓存非常有用;然而它的读写延迟却都翻倍了,严重影响了性能.下面探讨主要解决这些问题的缓存优化方法.

文献[27]提出了缓存行并行(line pairing)和行交换(line swapping)的策略来减少多层存储单元读写缓慢的问题.例如现有的2位存储单元包含一个硬位(hard-bit)和一个软位(soft-bit),hard-bit具有读快写慢的特点,而soft-bit具有读慢写快的特点,缓存line pairing策略用于组织和管理这2个位,而缓存line swapping策略用于将频繁写的数据放入soft-bit、将频繁读的数据放入hard-bit,这样就极大地降低了缓存的访问延迟和提高了整体的性能.如果soft-bit和hard-bit中的数据频繁地交换会带来一定的开销.

文献[39]提出了一种组重映射(set remapping)的方法来延长多层存储单元的使用时间.该方法每隔一段时间重新映射所有的组,改变数据存储的地址,任何2个不同的地址不会被映射到同一个块,组中任何位置都会被映射到,经过set remapping处理,所有缓存组耗损均衡了,提高了缓存的寿命.然而重新映射缓存会丢失性能,导致某些地址访问缺失,为了解决这个问题,该文献通过增加标志位来记录set remapping前后的地址变换,这样即可定位到映射前的数据.上述方法虽然延长了存储单元的使用时间,但带来了set remapping开销和存储空间的开销.

2.2.9 在GPU中的应用

近年来,在NVIDIA和AMD的推动下,GPU技术得到了迅猛发展,GPU中包含大量的计算核心,片上缓存的访问消耗了大量功耗,新型存储技术的引入能大幅度节省能耗.下面探讨一些主要的STT-RAM在GPU中应用的缓存优化方法.

文献[40]提出了在GPU中使用SRAM/STT-RAM共享缓存的方法来减少漏电功耗和动态功耗.由于不同的应用程序访问共享缓存表现出不同的时间局部性和空间局部性,对于具有良好时间局部性且写强度高的程序访问SRAM,其他的访问STT-RAM.GPU的共享缓存是软件管理方式的,用户可以根据程序的特点控制重复的数据访问,为了减少在STT-RAM中的重复写操作,采用了先读后写的方式,同时通过编译器辅助的方式指导数据的分配.上述方法对于减少GPU的功耗有良好的效果,对新型缓存技术在GPU中的应用有很大的参考价值.

文献[62]提出了在GPU中使用STT-RAM做二级缓存的架构方法.该缓存架构由小容量低保留时间数组(low-retention array, LR)和大容量高保留时间数组(high-retention array, HR)构成,LR负责存储和保留写强度高、时间局部性好的数据,而HR则用于存储只读或写不频繁的数据.为了最大化地利用二级缓存,对于HR中出现的频繁写操作应该迁移到LR中,通过写操作的上限次数来决定是否发生迁移.上述方法将不同保留时间的STT-RAM混合在一起,利用2部分缓存的特点分别处理GPU的读写操作,提高了性能,降低了功耗,同时也带来了刷新和迁移开销.

2.3 基于 PCM 的缓存优化方法

2.3.1 概 述

PCM 具有存储密度大、非易失性和可靠性高等优点;然而它的写速度慢,寿命也比 STT-RAM 低很多.少数研究者探讨了 PCM 在缓存中的应用,表 4 展示了现有的缓存优化方法,这些方法重点解决 PCM 存储寿命短的问题,主要从耗损均衡的角度均匀化写操作.

Table 4 Classification of Cache Optimization Approaches Based on PCM

表 4 基于 PCM 的缓存优化方法分类

Criterion	Approaches and References
Improving Cache Lifetime	Improving Lifetime[17]
	Reduce Inter and Intra-Set Write[33-34]
	Design[72]
Hybrid Caches	Architecture[6,71,73]
	Improving Lifetime[17]
Apply to GPU	Software-Hardware Co-Design[70]

2.3.2 基于 PCM 的主要缓存优化方法讨论

文献[72]提出了多种 PCM 的片上缓存架构优化技术,包括先读后写、数据反转编码和耗损均衡的方法.在缓存中,大约有 85%的写操作是重复的,为了减少这些重复的写操作,先读后写方法是利用 PCM 的读代价小于写代价的原理,将待写的新数据与读出的数据比较,若不同则写入新数据;为了进一步减少写操作的次数,数据反转编码方法是指新数据写入缓存块前先计算新数据和缓存块的汉明距离(hamming distance,HD),如果 HD 大于缓存块一半的容量,那么就将新数据反转编码后写入缓存块,并将反转状态位设为 1,下次根据反转状态位来恢复原始数据;为了使缓存块中写均匀分布,耗损均衡方法使用位行(bit line)移动的方式变换频繁访问数据的位置,通过移动偏移量来记录块的位置.上述方法极大地减少了写功耗,并将写寿命延长到了 3.8 年,然而反转编码会带来额外的时间和空间开销.

文献[73]提出了一种 SRAM/PCM 混合缓存架构的耗损均衡方法.该方法在缓存中加入读写计数器来记录数据的访问行为,并将数据分为写频繁和死写数据,最后根据这些信息预测数据的访问行为,并将其分配或移动到 SRAM 缓存中.上述方法能够避免在 PCM 中执行大量的写操作,降低系统功耗;然而小容量的 SRAM 对于写频繁的程序会造成访问冲突和等待,降低系统性能.

2.4 基于 RRAM 的缓存优化方法

RRAM 也具有存储密度大、非易失性和可靠性高等特点,它的读写速度要高于 PCM,发展前景最为广阔,然而它也存在寿命低的问题.表 5 介绍了 RRAM 在缓存中使用的优化方法.文献[33-34]通过均衡缓存组内和组间的写操作提高寿命.文献[34]提出了一种使用缓存数据迁移的技术来使组内写均衡的方法.该方法思想是如果写操作一直频繁发生在缓存组的 A 位置,那么隔一段时间就将其移动到缓存组中不频繁访问的 B 处,然后将 A 位置标记为无效,这样就达到了耗损均衡并提高缓存寿命的目的.上述方法会引起缓存访问缺失,降低性能.

Table 5 Classification of Cache Optimization Approaches Based on RRAM

表 5 基于 RRAM 的缓存优化方法分类

Criterion	Approaches and References
Improving Cache Lifetime	Reduce Inter and Intra-Set Write[33-34]

2.5 基于 DWM 的缓存优化方法

2.5.1 概 述

DWM 具有存储密度大、功耗低和非易失性等优点,存储寿命比 PCM 和 RRAM 都长,并且制造成本低,是极具吸引力的存储技术;然而它要面对位访问移动操作引起的写延迟高的问题.少数研究者在探索 DWM 在缓存中的应用,表 6 给出了 DWM 在缓存中使用的优化方法,主要从缓存架构、设计和减少写延迟等角度进行探讨.

Table 6 Classification of Cache Optimization Approaches Based on DWM

表 6 基于 DWM 的缓存优化方法分类

Criterion	Approaches and References
Design Exploration	Cross-Layer Design[74,77]
	Architecture[78]
Hybrid Caches	Architecture[9,11]
Using Multi-Level Cell	Shift Based Write[75]
	Architecture[76]
Apply to GPU	Architecture[79]
	Register File Architecture[80]

2.5.2 基于 DWM 的主要缓存优化方法讨论

文献[9]第一次提出一种基于 DWM 的 Tape-Cache 架构方法.该架构由 2 部分组成:1)针对读操作优化的多端口 DWM 宏单元;2)缓存宏单元的组织和管理策略.多端口宏单元从架构上看像磁带

(tape)结构,它可以存储高达数百个位,然而访问这些位所需的时间与读写端口的相对位置有关,这样就产生了不同的访问延迟.为了减少读访问延迟,TapeCache 宏单元将头部设计为只读端口,转换控制器能够迅速找到该单元.为了提高性能,DWM 的缓存块采用 DWM/SRAM 的混合结构,SRAM 做标志(tag)部分,DWM 做数组(array)部分.为了减少 TapeCache 访问的高延迟,该文献提出了多个 DWM 宏单元层间数据位组织方式,即缓存块的数据选取多个宏单元同一个位置数据,这样减少了访问单个宏单元 data array 时每个位访问延迟不同的弊端.上述方法为 DWM 的应用提供了一种架构思路,能有效地提高访问速度.

文献[77]提出了一种使用 DWM 架构最后一级缓存的交叉层(cross-layer)优化方法,包括缓存单元设计、数组结构、组织架构及数据管理等层面.缓存单元和数组的设计是为了使读写访问操作一致;组织架构方面采用了物理到逻辑的映射方法;数据管理方面使用了应用驱动的管理方式,即将访问频繁的数据块分配到离统一访问端口近的位置,这样可以最小化移动操作同时也减小了开销.上述方法从缓存的多个层面探索了 DWM 应用的潜力,为促进 DWM 在缓存中的应用具有十分重要的意义.

文献[79]第 1 次提出了一种基于 DWM 的 GPGPU 缓存层次结构设计方法.该方法在缓存的所有层次结构中都采用 1 位 DWM 和多位 DWM 组合的方式,充分地利用 DWM 的高密度性.访问数据的移动操作仍然是个挑战,因此将缓存块设计为成群的磁带间并行组织方式,这样磁带可以共享一个控制逻辑并且缓存块的各个位访问速度一致.为了减少连续访问操作的移动延迟,该架构设计了磁带头部管理策略,即磁带头部的读写端口是对齐的.为了进一步减少访问延迟,该架构在 L2 缓存前增加了一个移动感知的缓冲区,当 GPGPU 中的不同流处理器访问 L2 时,如果预测到该访问具有内部 warp 局部性或表现出高的移动代价,那么该访问将转移到缓存区.上述 4 种方案综合降低了访问延迟,降低功耗并能提升性能.

3 新型 NVM 架构的缓存优化方法展望

目前,新型非易失性存储技术在体系结构中的商业应用非常广泛,学术界已经从存储系统的多个角度进行了探讨,并取得了显著的成果.随着大数据、云计算和物联网时代的到来,应用场景希望缓存系统的体积小、访问速度快和稳定性高等,传统缓存系统发展已经遇到瓶颈,新型 NVM 展现出的优良特性为解决应用场景的需求提供了可能.展望未来,新型 NVM 的广泛应用仍有许多挑战性的问题需要不断地探索和研究,主要表现在以下 4 个方面:

1) 基于新型 NVM 的缓存体系结构优化方法

当前的缓存系统架构主要由 SRAM 构成,它有读写速度一致、存储容量小和使用寿命长等特点;而新型 NVM 读写不对称、写功耗大和寿命有限等问题不适合直接应用于当前缓存体系结构,需要从电路级、系统级和架构级针对性地设计,如针对写操作和写功耗优化、针对耗损均衡优化和针对性能的优化等.为解决单一 NVM 存储介质的缺点,可以构建同构混合缓存,如不同保留时间的 STT-RAM 组合;构建异构混合缓存,如 SRAM 与 STT-RAM、STT-RAM 与 DRAM 等;构建多层次混合缓存,如 SRAM 做 L1,STT-RAM 做 L2 和 PCM 做 L3 等,通过混合缓存方式充分发挥多种存储介质的优势,最终形成最优的新型缓存体系结构,这是值得深入研究的问题.

2) 基于新型 NVM 架构的缓存软件优化方法

由于新型 NVM 存储特性不同于传统存储介质,现有软件层的方法需进一步改进和优化.在缓存管理方面,需要针对新型 NVM 设计新的缓存数据结构和优化算法,减少不必要的写操作;在程序执行前,可以采用编译优化技术指导数据的分配和迁移,提升系统性能;在不同的应用场景,预测程序读写访问特点,动态地调整缓存访问策略,实现耗损均衡;在多核环境中,优化数据分配策略,使多核私有缓存访问均衡;在线程管理方面,研究线程调度算法优化,减少线程间访问冲突,线程访问均匀化,从而提升缓存系统性能;在缓存存储层次,优化各个层次间的带宽,增大数据吞吐率,缩短访问缓存的时间,加速系统的处理.软件层面的优化方法研究直接影响硬件资源的最大化及最优化利用,因此这是一个非常有应用价值的研究方向.

3) 基于新型 NVM 架构的缓存数据可靠性优化方法

可靠性对于系统具有非常重要的意义,一旦数据损坏或丢失将会造成不可估量的后果.随着新型 NVM 制造工艺的不断提升,它们的存储单元越来越小,容量越来越大;然而存储单元的错误率也越来

越高,同时新型 NVM 的写次数有限,如 PCM 和 RRAM 写的次数在 10^9 左右,当写操作达到极限后,存储芯片就会坏损,这给新型 NVM 中存储的数据可靠性带来了挑战.

为了保证缓存数据的可靠性,可以研究新的耗损均衡算法,根据应用场景的不同动态地调整读写策略,延长新型 NVM 的使用时间;研究新的缓存访问控制算法来减少写操作;研究低开销的纠错方法,在保证纠错准确率的前提下尽量减小开销;研究坏块复用方法,将坏块中未损坏的部分组合起来;研究坏块丢弃策略,将缓存中出现的坏块丢弃,然后指导其他数据正确分配和访问并维护数据的一致性. 缓存数据可靠性研究是新型 NVM 在缓存架构中的重点,是需要深入研究和探索的.

4) 新型 NVM 在 GPU 缓存架构中的优化方法

近年来,应用场景对计算需求越来越高,GPU 的应用逐渐流行起来,它具有浮点计算能力强、带宽高和性价比高等优点;然而随着半导体工艺的发展,GPU 中 warp 数量越来越多,对存储容量的需求也与日俱增. 传统存储技术的应用遇到瓶颈,新型 NVM 可以有效解决容量的问题,为此,研究新型 NVM 在 GPU 缓存架构中的优化方法是一个非常有潜力的研究方向.

未来的研究可能包括以下 4 方面:1)研究在 GPU 中引入新型 NVM 并设计新的缓存体系结构,从存储单元、电路级、系统级和架构等方面设计和优化;2)研究 GPU 下新型缓存的数据管理方法,充分利用新型 NVM 的优点并克服其缺点;3)研究在 GPU 局部缓存、共享缓存和全局缓存中使用新型 NVM 或混合 NVM;4)研究 GPU 缓存层次间的带宽和网络负载优化,提高访问速度并降低功耗.

综上所述,考虑到新时代对计算速度、存储容量、系统功耗和性能等综合因素的需求,在新缓存体系结构下,如何充分发挥新型 NVM 的优势,通过各种优化方法弥补它们的缺陷,最大限度地降低功耗并提高性能,并为大数据、云计算和物联网环境服务,这是未来的发展趋势和重点研究方向.

4 结束语

新型 NVM 具有非易失性、低功耗和高存储密度等优点,对传统存储体系结构将产生深远影响,然而它也存在着写功耗高、写寿命有限和写延迟长等缺点. 本文总结了新型 NVM 的特点、新型 NVM 架构的缓存优化方法和新型 NVM 在未来缓存架构中优化方法的研究方向;重点分类探讨了主流的缓存优化方法,分析了新型 NVM 应用于缓存系统优化方法的优势和缺陷,这些有助于研究者和工程师运用以上优化方法提高缓存的性能和能效,同时设计新的解决方案来优化缓存,为新型 NVM 广泛使用提供了一种思路.

新型存储器件能为下一代高性能计算提供高存储密度、为大数据提供低能耗和快速访问的存储设施、为物联网提供非易失性低能耗的存储器和为移动终端提供大容量高速度的存储设备,这些特性给计算机系统结构和软件设计带来了巨大的机遇. 目前,新型 NVM 在体系结构中的商业应用非常广泛,STT-RAM 和 PCM 已经出现了相关产品,但存在着各类问题需要逐一解决. 随着新型存储器件的制造工艺逐渐成熟、性能更加稳定、写功耗更低及使用时间更长,那么新型存储器的应用将会普及,新的缓存体系结构和系统软件优化也将得到空前的发展. 这些都是未来需要进一步深入研究和探索的.

参 考 文 献

[1] Intel. Intel® Xeon® Processor E5-2699 v3 [OL]. [2014-12-15]. http://ark.intel.com/products/81061

[2] Vardhan A, Srikant Y. Exploiting critical data regions to reduce data cache energy consumption [C] //Proc of the 17th Int Workshop on Software and Compilers for Embedded Systems. New York: ACM, 2013: 69-78

[3] Shen Zhirong, Xue Wei, Shu Jiwu. Research on new non-volatile storage [J]. 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[J]. 计算机研究与发展, 2014, 51(2): 445-453)

[4] Zhang Hongbin, Fan Jie, Shu Jiwu, et al. Summary of storage system and techmology based on phase change memory [J]. Journal of Computer Research and Development, 2014, 51(8): 1647-1662 (in Chinese)
(张鸿斌, 范捷, 舒继武, 等. 基于相变存储器的存储系统与技术综述[J]. 计算机研究与发展, 2014, 51(8): 1647-1662)

[5] 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 [C] //Proc of the 19th Int Symp on High Performance Computer Architecture (HPCA2013). Piscataway, NJ: IEEE, 2013: 143-154

[6] Wu Xiaoxia, Li Jian, Zhang Lixin, et al. Hybrid cache architecture with disparate memory technologies [C] //Proc of the 36th Annual Int Symp on Computer Architecture (ISCA 2009). New York: ACM, 2009: 34-45

[7] Chen Yuting, Cong Jason, Huang Hui, et al. Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design [C] //Proc of the Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2012: 12-16

[8] Li Yong, Chen Yiran, Jones A. A software approach for combating asymmetries of non-volatile memories [C] //Proc of the 2012 ACM/IEEE Int Symp on Low Power Electronics and Design (ISLPED 2012). New York: ACM, 2012: 191-196

[9] Venkatesan R, Kozhikkottu V, Augustine C, et al. TapeCache: A high density, energy efficient cache based on domain wall memory [C] //Proc of the 2012 ACM/IEEE Int Symp on Low Power Electronics and Design (ISLPED 2012). New York: ACM, 2012: 185-190

[10] Kryder M, Chang S. After hard drives—What comes next? [J]. IEEE Trans on Magnetics, 2009, 10(45): 3406-3413

[11] Zhao W, Zhang Y, Trinh H, et al. Magnetic domain-wall racetrack memory for high density and fast data storage [C] //Proc of the 11th IEEE Int Conf on Solid-State and Integrated Circuit Technology (ICSICT2012). Piscataway, NJ: IEEE, 2012: 1-4

[12] Smullen C, Mohan V, Nigam A, et al. Relaxing non-volatility for fast and energy-efficient STT-RAM caches [C] //Proc of the 17th IEEE Int Symp on High Performance Computer Architecture (HPCA2011). Piscataway, NJ: IEEE, 2011: 50-61

[13] Zhou Ping, Zhao Bo, Yang Jun, et al. Energy reduction for STT-RAM using early write termination [C] //Proc of the 2009 IEEE/ACM Int Conf on Computer-Aided Design-Digest of Technical Papers. Piscataway, NJ: IEEE, 2009: 264-268

[14] Li Hai, Chen Yiran. An overview of non-volatile memory technology and the implication for tools and architectures [C] //Proc of 2009 Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2009: 731-736

[15] Li Yong, Zhang Yaojun, Li Hai, et al. C1C: A configurable, compiler-guided STT-RAM L1 cache [J]. ACM Trans on Architecture and Code Optimization (TACO), 2013, 10(4): 1-22

[16] Sun Zhenyu, Li Hai, Wu Wenqing. A dual-mode architecture for fast-switching STT-RAM [C] //Proc of the 2012 ACM/IEEE Int Symp on Low Power Electronics and Design. New York: ACM, 2012: 45-50

[17] Joo Y, Park S. A hybrid PRAM and STT-RAM cache architecture for extending the lifetime of PRAM caches [J]. IEEE Computer Architecture Letters, 2013, 12(2): 55-58

[18] Sun Guangyu, Dong Xiangyu, Xie Yuan, et al. A novel architecture of the 3D stacked MRAM L2 cache for CMPs [C] //Proc of the 15th IEEE Int Symp on High Performance Computer Architecture (HPCA2009). Piscataway, NJ: IEEE, 2009: 239-249

[19] Rasquinha M, Choudhary D, Chatterjee S, et al. An energy efficient cache design using spin torque transfer (STT) RAM [C] //Proc of the 16th ACM/IEEE Int Symp on Low Power Electronics and Design. New York: ACM, 2010: 389-394

[20] Kwon K W, Choday S H, Kim Y, et al. AWARE (asymmetric write architecture with REdundant blocks): A high write speed STT-MRAM cache architecture [J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2014, 22(4): 712-720

[21] Zhou Yi, Zhang Chao, Sun Guangyu, et al. Asymmetric-access aware optimization for STT-RAM caches with process variations [C] //Proc of the 23rd ACM Int Conf on Great Lakes Symp on VLSI. New York: ACM, 2013: 143-148

[22] Bishnoi R, Ebrahimi M, Oboril F, et al. Asynchronous asymmetrical write termination (AAWT) for a low power STT-MRAM [C] //Proc of Design, Automation and Test in Europe Conf and Exhibition (DATE2014). Piscataway, NJ: IEEE, 2014: 1-6

[23] Bishnoi R, Oboril F, Ebrahimi M, et al. Avoiding unnecessary write operations in STT-MRAM for low power implementation [C] //Proc of the 15th Int Symp on Quality Electronic Design (ISQEDv). Piscataway, NJ: IEEE, 2014: 548-553

[24] Noguchi H, Nomura K, Abe K, et al. D-MRAM cache: Enhancing energy efficiency with 3T-1MTJ DRAM/MRAM hybrid memory [C] //Proc of the Int Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2013: 1813-1818

[25] Jog A, Mishra A K, Xu C, et al. Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs [C] //Proc of the 49th Annual Design Automation Conf. New York: ACM, 2012: 243-252

[26] Yazdanshenas S, Pirbast M R, Fazeli M, et al. Coding last level STT-RAM cache for high endurance and low power [J]. IEEE Computer Architecture Letters, 2013, 13(2): 73-76

[27] Jiang Lei, Zhao Bo, Zhang Youtao, et al. Constructing large and fast multi-level cell STT-MRAM based cache for embedded processors [C] //Proc of the 49th ACM/EDAC/IEEE Design Automation Conf (DAC2012). Piscataway, NJ: IEEE, 2012: 907-912

[28] Mao Mengjie, Li Hai, Jones A K, et al. Coordinating prefetching and STT-RAM based last-level cache management for multicore systems [C] //Proc of the 23rd ACM Int Conf on Great Lakes Symp on VLSI. New York: ACM, 2013: 55-60

[29] Li Yong, Jones A K. Cross-layer techniques for optimizing systems utilizing memories with asymmetric access characteristics [C] //Proc of 2012 IEEE Computer Society Annual Symp on VLSI (ISVLSI). Piscataway, NJ: IEEE, 2012: 404-409

[30] Jung J, Nakata Y, Yoshimoto M, et al. Energy-efficient spin-transfer torque RAM cache exploiting additional all-zero-data flags [C] //Proc of the 14th Int Symp on IEEE Quality Electronic Design (ISQED2013). Piscataway, NJ: IEEE, 2013: 216-222

[31] Park S P, Gupta S, Mojumder N, et al. Future cache design using STT MRAMs for improved energy efficiency: Devices, circuits and architecture [C] //Proc of the 49th Annual Design Automation Conf. New York: ACM, 2012: 492-497

[32] Jadidi A, Arjomand M, Sarbazi-Azad H. High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement [C] //Proc of the 17th IEEE/ACM Int Symp on Low-Power Electronics and Design. Piscataway, NJ: IEEE, 2011: 79-84

[33] Wang Jue, Dong Xiangyu, Xie Yuan, et al. i^{2}WAP: Improving non-volatile cache lifetime by reducing inter-and intra-set write variations [C] //Proc of the 19th IEEE Int Symp on High Performance Computer Architecture (HPCA2013). Piscataway, NJ: IEEE, 2013: 234-245

[34] Mittal S, Vetter J S, Li D. LastingNVCache: A technique for improving the lifetime of non-volatile caches [C] //Proc of 2014 IEEE Computer Society Annual Symp on VLSI (ISVLSI). Piscataway, NJ: IEEE, 2014: 534-540

[35] Ahn J, Choi K. LASIC: Loop-aware sleepy instruction caches based on STT-RAM Technology [J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2014, 22(5): 1197-1201

[36] Strikos N, Kontorinis V, Dong X, et al. Low-current probabilistic writes for power-efficient STT-RAM caches [C] //Proc of the 31st IEEE Int Conf on Computer Design (ICCD2013). Piscataway, NJ: IEEE, 2013: 511-514

[37] Ahn J, Choi K. Lower-bits cache for low power STT-RAM caches [C] //Proc of 2012 IEEE Int Symp on Circuits and Systems (ISCAS). Piscataway, NJ: IEEE, 2012: 480-483

[38] Sun Zhenyu, Bi Xiuyuan, Li Hai, et al. Multi retention level STT-RAM cache designs with a dynamic refresh scheme [C] //Proc of the 44th Annual IEEE/ACM Int Symp on Microarchitecture. New York: ACM, 2011: 329-338

[39] Chen Yiran, Wong Wengfai, Li Hai, et al. On-chip caches built on multilevel spin-transfer torque RAM cells and its optimizations [J]. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2013, 9(2): 1-22

[40] Goswami N, Cao B, Li T. Power-performance co-optimization of throughput core architecture using resistive memory [C] //Proc of the 19th IEEE Int Symp on High Performance Computer Architecture (HPCA2013). Piscataway, NJ: IEEE, 2013: 342-353

[41] Quan Baixing, Zhang Tiefei, Chen Tianzhou, et al. Prediction table based management policy for STT-RAM and SRAM hybrid cache [C] //Proc of the 7th Int Conf on Computing and Convergence Technology (ICCCT2012). Piscataway, NJ: IEEE, 2012: 1092-1097

[42] Bi Xiuyuan, Sun Zhenyu, Li Hai, et al. Probabilistic design methodology to improve run-time stability and performance of stt-ram caches [C] //Proc of the 2012 IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2012: 88-94

[43] Guo Xiaochen, Ipek E, Soyata T. Resistive computation: Avoiding the power wall with low-leakage, STT-MRAM based computing [C] //Proc of ACM SIGARCH Computer Architecture News. New York: ACM, 2010: 371-382

[44] Ahn J, Yoo S, Choi K. Selectively protecting error-correcting code for area-efficient and reliable STT-RAM caches [C] //Proc of the 18th Asia and South Pacific Design Automation Conf (ASP-DAC2013). Piscataway, NJ: IEEE, 2013: 285-290

[45] Chen Yuting, Cong Jason, Huang Hui, et al. Static and dynamic co-optimizations for blocks mapping in hybrid caches [C] //Proc of the 2012 ACM/IEEE Int Symp on Low Power Electronics and Design. New York: ACM, 2012: 237-242

[46] Satyamoorthy P. STT-RAM for shared memory in GPUs [D]. Charlattesville, VA: University of Virginia, 2011

[47] Bi Xiuyuan, Mao Mengjie, Wang Danghui, et al. Unleashing the potential of MLC STT-RAM caches [C] //Proc of the 2013 IEEE/ACM Int Conf on Computer-Aided Design. Piscataway, NJ: IEEE, 2013: 429-436

[48] Mittal S. Using cache-coloring to mitigate inter-set write variation in non-volatile caches [J/OL]. CoRR, 2013, abs/1310.8494. [2014/12/15]. http://arxiv.org/abs/1310.8494

[49] Sun Hongbin, Liu Chuanyin, Xu Wei, et al. Using magnetic RAM to build low-power and soft error-resilient L1 cache [J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2012, 20(1): 19-28

[50] Ahn J, Yoo S, Choi K. Write intensity prediction for energy-efficient non-volatile caches [C] //Proc of 2013 IEEE Int Symp on Low Power Electronics and Design (ISLPED). Piscataway, NJ: IEEE, 2013: 223-228

[51] Syu S M, Shao Y H, Lin I C. High-endurance hybrid cache design in CMP architecture with cache partitioning and access-aware policy [C] //Proc of the 23rd ACM Int Conf on Great Lakes Symp on VLSI. New York: ACM, 2013: 19-24

[52] Li Qingan, Li Jianhua, Shi Liang, et al. Compiler-assisted refresh minimization for volatile STT-RAM cache [C] //Proc of the 18th Asia and South Pacific Design Automation Conf (ASP-DAC2013). Piscataway, NJ: IEEE, 2013: 273-278

[53] Li Jianhua, Shi Liang, Li Qingan, et al. Cache coherence enabled adaptive refresh for volatile STT-RAM [C] //Proc of the Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2013: 1247-1250

[54] Li Qingan, Zhao Mengying, Xue Chun, et al. Compiler-assisted preferred caching for embedded systems with STT-RAM based hybrid cache [C] //Proc of the 13th ACM SIGPLAN/SIGBED Int Conf on Languages, Compilers, Tools and Theory for Embedded Systems. New York: ACM, 2012: 109-118

[55] Li Qingan, Li Jianhua, Shi Liang, et al. Compiler-assisted STT-RAM-based hybrid cache for energy efficient embedded systems [J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2014, 22(8): 1829-1840

[56] Li Jianhua, Shi Liang, Li Qingan, et al. Thread progress aware coherence adaption for hybrid cache coherence protocols [J]. IEEE Trans on Parallel and Distributed Systems, 2013, 25(10): 2697-2707

[57] Li Jianhua, Shi Liang, Li Qingan, et al. Low-energy volatile STT-RAM cache design using cache-coherence-enabled adaptive refresh [J]. ACM Trans on Design Automation of Electronic Systems (TODAES), 2013, 19(1): 1-23

[58] Wang Jianxing, Tim Y, Wong Wengfai, et al. A coherent hybrid SRAM and STT-RAM L1 cache architecture for shared memory multicores [C] //Proc of the 19th Asia and South Pacific Design Automation Conf (ASP-DAC2014). Piscataway, NJ: IEEE, 2014: 610-615

[59] Ahn J, Yoo S, Choi K. Dasca: Dead write prediction assisted STT-RAM cache architecture [C] //Proc of the 20th IEEE Int Symp on High Performance Computer Architecture (HPCA2014). Piscataway, NJ: IEEE, 2014: 25-36

[60] Wang Zhe, Jiménez D A, Xu Cong, et al. Adaptive placement and migration policy for an STT-RAM-based hybrid cache [C] //Proc of the 20th IEEE Int Symp on High Performance Computer Architecture (HPCA2014). Piscataway, NJ: IEEE, 2014: 13-24

[61] Zhao Jishen, Xie Yuan. Optimizing bandwidth and power of graphics memory with hybrid memory technologies and adaptive data migration [C] //Proc of the 2012 IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2012: 81-87

[62] Samavatian M H, Abbasitabar H, Arjomand M, et al. An efficient STT-RAM last level cache architecture for GPUs [C] //Proc of the 51st ACM Design Automation Conf. New York: ACM, 2014: 1-6

[63] Liu Xiaoxiao, Li Yong, Zhang Yaojun, et al. STD-TLB: A STT-RAM-based dynamically-configurable translation lookaside buffer for GPU architectures [C] //Proc of the 19th Asia and South Pacific Design Automation Conf (ASP-DAC2014). Piscataway, NJ: IEEE, 2014: 355-360

[64] Wei Wei, Jiang Dejun, Xiong Jin, et al. HAP: Hybrid-memory-aware partition in shared last-level cache [C] //Proc of the 32nd IEEE Int Conf on Computer Design (ICCD2014). Piscataway, NJ: IEEE, 2014: 28-35

[65] Dong Xiangyu, Wu Xiaoxia, Sun Guangyu, et al. Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement [C] //Proc of the 45th ACM/IEEE Design Automation Conf (DAC2008). Piscataway, NJ: IEEE, 2008: 554-559

[66] Wang Jue, Dong Xiangyu, Xie Yuan. OAP: An obstruction-aware cache management policy for STT-RAM last-level caches [C] //Proc of the Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2013: 847-852

[67] Mao Mengjie, Sun Guangyu, Li Yong, et al. Prefetching techniques for STT-RAM based last-level cache in CMP systems [C] //Proc of the 19th Asia and South Pacific Design Automation Conf (ASP-DAC2014). Piscataway, NJ: IEEE, 2014: 67-72

[68] Sun Guangyu, Zhang Yaojun, Wang Yu, et al. Improving energy efficiency of write-asymmetric memories by log style write [C] //Proc of the 2012 ACM/IEEE Int Symp on Low Power Electronics and Design. New York: ACM, 2012: 173-178

[69] Zhang Yaojun, Bayram I, Wang Yu, et al. ADAMS: Asymmetric differential STT-RAM cell structure for reliable and high-performance applications [C] //Proc of the 2013 IEEE/ACM Int Conf on Computer-Aided Design. Piscataway, NJ: IEEE, 2013: 9-16

[70] Wang Bin, Wu Bo, Li Dong, et al. Exploring hybrid memory for GPU energy efficiency through software-hardware co-design [C] //Proc of the 22nd Int Conf on Parallel Architectures and Compilation Techniques. Piscataway, NJ: IEEE, 2013: 93-102

[71] Mangalagiri P, Sarpatwari K, Yanamandra A, et al. A low-power phase change memory based hybrid cache architecture [C] //Proc of the 18th ACM Great Lakes Symp on VLSI. New York: ACM, 2008: 395-398

[72] Joo Y, Niu D, Dong X, et al. Energy-and endurance-aware design of phase change memory caches [C] //Proc of the Conf on Design, Automation and Test in Europe (DATE). Piscataway, NJ: IEEE, 2010: 136-141

[73] Guo Sanchuan, Liu Zhenyu, Wang Dongsheng, et al. Wear-resistant hybrid cache architecture with phase change memory [C] //Proc of the 7th IEEE Int Conf on Networking, Architecture and Storage (NAS2012). Piscataway, NJ: IEEE, 2012: 268-272

[74] Sun Zhenyu, Wu Wenqing, Li Hai. Cross-layer racetrack memory design for ultra high density and low power consumption [C] //Proc of the 50th ACM/EDAC/IEEE Design Automation Conf (DAC2013). Piscataway, NJ: IEEE, 2013: 1-6

[75] Venkatesan R, Sharad M, Roy K, et al. DWM-TAPESTRI-an energy efficient all-spin cache using domain wall shift based writes [C] //Proc of the Conf on Design, Automation and Test in Europe (DATE 2013). Piscataway, NJ: IEEE, 2013: 1825-1830

[76] Sharad M, Venkatesan R, Raghunathan A, et al. Multi-level magnetic RAM using domain wall shift for energy-efficient, high-density caches [C] //Proc of 2013 IEEE Int Symp on Low Power Electronics and Design (ISLPED). Piscataway, NJ: IEEE, 2013: 64-69

[77] Sun Zhenyu, Bi Xiuyuan, Wu Wenqing, et al. Array organization and data management exploration in racetrack memory [J]. IEEE Trans on Computers, 2014, 13(6): 1-14

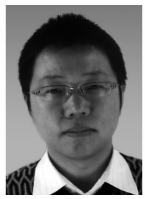
[78] Venkatesan R, Chippa V K, Augustine C, et al. Domain-specific many-core computing using spin-based memory [J]. IEEE Trans on Nanotechnology, 2014, 13(5): 881-894

[79] Venkatesan R, Ramasubramanian S G, Venkataramani S, et al. STAG: Spintronic-tape architecture for GPGPU cache hierarchies [C] //Proc of the 41st ACM/IEEE Int Symp on Computer Architecture (ISCA2014). Piscataway, NJ: IEEE, 2014: 253-264

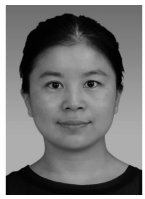
[80] Mao M, Wen W, Zhang Y, et al. Exploration of GPGPU register file architecture using domain-wall-shift-write based racetrack memory [C] //Proc of the 51st Annual Design Automation Conf. New York: ACM, 2014: 1-6

He Yanxiang, born in 1952. PhD, professor and PhD supervisor. Senior member of China Computer Federation. His main research interests include compiler theory, trusted software, distributed parallel processing and software engineering.

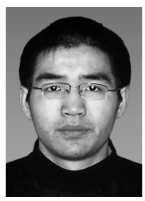
Shen Fanfan, born in 1987. PhD candidate. His main research interests include emerging non-volatile memory, embedded system and computer architecture.



Zhang Jun, born in 1978. PhD candidate and associate professor. Member of China Computer Federation. His main research interests include trusted software and high performance computing (zhangjun_whu@whu.edu.cn).



Jiang Nan, born in 1976. PhD candidate and lecturer. Member of China Computer Federation. Her main research interests include trusted software and trusted compiler(nanjiang@whu.edu.cn).



Li Qing'an, born in 1986. PhD and lecturer. Member of China Computer Federation. His main research interests include compiler optimization and embedded system(qali@whu.edu.cn).

Li Jianhua, born in 1985. PhD and lecturer. His main research interests include on-chip networks, emerging non-volatile memory and embedded system(jhli@hfut.edu.cn).