面向大数据应用的混合内存架构特征分析

李鑫[1]，陈璇[2]，黄志球[1]
1. 南京航空航天大学计算机科学与技术学院，江苏 南京 211106；
2. 南京航空航天大学自动化学院，江苏 南京 211106

摘要
受限于DRAM的扩展性，大数据分析及相关应用性能难以有效提升。新型非易失性存储器凭借其非易失性、高存储密度、低能耗等优点，为大数据应用的性能与效率提升带来了契机。以新型非易失性存储器为基础，阐述PCM/DRAM混合存储架构，通过对该混合存储架构在性能优化、能耗优化、内存管理策略等方面的综述分析，详述了混合存储架构在大数据应用方面的优势及可行性，总结了现有研究工作的缺陷，展望了PCM/DRAM混合内存后续的研究方向。

关键词
大数据；非易失性存储器；相变存储器；性能优化

中图分类号：TP302.7 文献标识码：A doi: 10.11959/j.issn.2096-0271.2018031

Analysis on hybrid memory architecture for big data application

LI Xin[1], CHEN Xuan[2], HUANG Zhiqiu[1]
1. College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
2. College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Abstract
Due to the limited scalability of DRAM, it is hard to optimize the performance of big data analysis and the big data applications. The new non-volatile memory (NVM) brings the opportunity to improve the performance and efficiency for big data applications, which benefits by the advantages of NVM, including its non-volatile, high storage density, and low power consumption. The PCM/DRAM hybrid memory architecture based on the non-volatile memory was analyzed. The feasibility and advantages of hybrid memory for big data applications through the analysis on the optimization of performance, energy consumption and memory management strategies for hybrid memory architecture were demonstrated. The defects in existing work were summarized and the potential research field in PCM/DRAM hybrid memory architecture was discussed.

Key words
big data, non-volatile memory, phase change memory, performance optimization

1 引言

随着大数据的出现及大数据分析技术的发展，大数据应用受到越来越广泛的关注。大数据具有数据量巨大、数据种类繁多、数据价值密度低以及处理数据时效性要求高等特点[1]。大数据应用需要执行大量计算工作，同时对大数据的处理与存储也有着低时延、低开销、高效率等需求。现在无论是数据中心里的超级计算机还是个人计算机都利用以动态随机存取存储器（dynamic random access memory，DRAM）为核心构成的内存架构来管理和存储大数据，DRAM的可扩展性受限会增加大数据分析的操作时间，从而降低吞吐量，无法高效地对大数据进行存储和分析。虽然工业界和学术界一直都在软件方面研究并尝试解决这一系列的问题，并在一定程度上缓解了现有存储架构的缺陷，但却很难获得本质上的突破。

新型非易失性存储器（non-volatile memory，NVM）的出现，给传统的以DRAM为主体构成的内存系统带来了挑战，也为优化大数据应用提供了契机。其中，相变寄存器（phase change memory，PCM）被认为是目前有可能取代DRAM作为内存构成的选择之一。与DRAM相比，PCM具有非易失性、高存储密度和良好扩展性等合乎大数据存储技术需求的特征。但是，非易失性存储器还存在以下问题。

- PCM读写不对称。在性能方面，写时延相对DRAM较长，会导致访问内存的时间延长，降低系统的性能；在能耗方面，对PCM进行写操作比读操作的能耗要高，会导致更多的能源消耗。
- PCM的耐写度有限。数据在PCM内存架构上的写操作分布不均匀会缩短PCM的寿命，也会对存储在PCM上的数据的安全性造成影响。

由此可见，如果用PCM完全取代DRAM作为构成计算系统的内存，会对计算系统的寿命、性能、能耗和安全性等造成一定的影响。因此，必须有效解决上述问题，才能发挥PCM在优化大数据应用方面的效用，而采用基于PCM和DRAM的混合内存架构是当前的主要方式。

本文从分析大数据应用和NVM的特征入手，旨在分析PCM/DRAM混合存储架构在优化大数据应用方面的可行性及优化方向。通过研究比较DRAM与PCM不同的组成方案和管理策略，从混合存储架构的性能优化和能耗优化两方面分析主要的优化算法和相关的故障处理，并讨论未来的优化方向，以达到最大限度地利用DRAM和PCM优势的目的，为全面利用PCM/DRAM混合内存架构开展大数据应用调度优化提供基础。

2 大数据应用及NVM的特征

2.1 典型应用场景下大数据应用特征

随着大数据概念的出现，学术界和工业界都利用大数据分析技术的优势开展应用，以提升服务或应用效率，现今大数据典型应用场景有：企业内部大数据应用、物联网大数据应用、面向在线社交网络大数据的应用、医疗健康大数据应用、群智感知和智能发电等[2]。这些应用体现了大数据的数字化、全球化、超海量、实时性、价值密度低等特点[3,4]。大数据的应用特征表现在以下两个方面。

（1）数据处理时效性要求高，处理速度问题突出

许多嵌入式的系统都会产生大量的物理数据，需要动态地处理分析这些数据。企业大数据应用也需要实时地对数据的变化做出应对和决策。数据处理的响应时间也从批处理响应时间逐渐转变为实时的流数据处理响应时间[5]。根据国际数据公司（International Data Corporation）发布的名为《大数据，更大的数字身影，最大增长在远东》的研究报告，预计到2020年，数字宇宙规模将达到40 ZB[6]。这些均表明大数据时代对数据处理效率有着迫切的需求。

（2）数据精确性要求高

数据来源的多元化降低了数据的可靠度和质量，但是面向大数据的计算系统需要追求高并发、高性能读写访问、低功耗等特性，其精确需求难以很好地满足。

2.2 大数据应用在传统存储架构下的瓶颈

大数据应用的特征使大数据处理存在很多困难，在传统存储架构下，计算机内存容量有限、输入/输出压力大等缺陷使大数据处理效率低、能耗高。大数据应用面临操作（分析、查询等）时延长、能源消耗大和存储容量有限这3个瓶颈。

（1）操作时延长

在传统的冯·诺伊曼结构中，CPU的处理速率远快于内存的处理速率，当CPU需要在大量的资源或数据上执行一些简单的指令时，由于I/O流量与CPU的工作效率相差太大，计算机运行的整体效率受到严重的限制。现实中，处理器和内存的性能一直在提升但却具有不同的提升速率，两者之间的带宽差距也在增加。大数据继承了互联网的数字化表示，传统的内存器件DRAM用电容的充放电来表示“0”和“1”，为了防止电容因漏电而导致信息丢失，需要周期性地刷新DRAM以保存DRAM中的数据，这就带来了计算系统的额外时间开销，导致大数据的实时性需求得不到满足。

（2）能源消耗大

能源消耗是现代计算系统设计的一个重要考虑因素。近年来，能源管理的研究大多集中在中央处理器的动态管理上，研究人认为它是能源消耗的最主要因素。然而，最近的研究表明，在现代计算系统中，内存已经成为最显著的能源消耗部件，占据能源总消耗的30%～50%[7-11]。

DRAM内存被组织为一个包含行和列的网格，每一位数据都以小电容充电的形式存储在这个网格中。漏电和频繁的访问会导致电荷耗尽，DRAM需要一个持续的刷新操作来维持它的数据，因此，进行刷新操作的电源就会导致持续的能源消耗。同时DRAM设置行和列给物理地址访问时要消耗能源。当其他行需要访问时，DRAM关闭一行也需要额外的能源开销。此外，在进行实际的读写操作时，因为漏电和周期性的供应，持续的备用电源都会造成能源的损耗。

虽然关键的大数据技术仍处在初步阶段[2]，但是学术界和工业界对大数据的应用已经越来越广泛，这些应用更多地转移到包含大量信息和通信技术的大数据中心，呈现大数据中心化的特征。目前大数据中心包括数以万计的服务器，其能源消耗量甚至可以超过一座小型城镇的能源消耗量[12]。与此同时，这些服务器在日常工作中约有30%的时间是不承担任何任务的，闲置的服务器只消耗能源，不产生价值，大数据中心的能源利用率普遍只有5%～10%[12]。

（3）存储容量有限

当前需要存储和处理的大数据达到了

PB量级，因此存储器的存储容量和存储密度也是一个亟须解决的问题。由于磁盘的I/O速度比计算系统其他部分慢5个数量级[13]，如果扩大磁盘容量，寻址时间会随磁盘容量的扩大而增加，进而增加操作的时延，从而降低I/O的吞吐量。由于DRAM存储密度较小、价格较高，如果扩大DRAM内存容量，则会导致能源消耗进一步加剧，并显著增加计算系统的成本。

学术界与工业界都尝试在软件方面对现有的存储机构进行改进，解决大数据存储的问题，其中包括以Hadoop分布式文件系统（Hadoop distributed file system，HDFS）[14]和以非关系型数据库（not only SQL，NoSQL）为代表的大规模分布式数据库系统设计、基于以DRAM为核心的内存数据库技术等。然而，这些软件或软硬件结合的方案都是从传统的DRAM内存架构考虑的，没有实质上的突破。在大数据应用的环境下，内存与外存之间的处理速率仍然相差很大，需要从硬件的角度考虑才能更好地满足大数据应用的需求。

2.3 新型非易失性存储器

由前文可知，以DRAM为核心构成内存的传统架构已经不能满足大数据的应用需求。随着新型的非易失性存储器阻变式存储器（resistive random access memory，RRAM）、铁电存储器（ferroelectric random access memory，FRAM）、磁阻内存（magnetic random access memory，MRAM）、相变存储器（phase change memory，PCM）以及闪存（flash memory）走出实验室，NVM成本降低并实现了产品化，为研究适合高效率、低能耗的大数据存储和管理的新型存储架构带来了新的机遇。

闪存技术的快速发展给数据管理研究带来了巨大冲击，但是受按页存取的方式和存取性能等因素限制，闪存较适合作为二级存储器[15]。基于闪存的数据存储与管理技术只是优化了磁盘级别的I/O时延[16]，对传统存储架构的变革没有太大的影响。

其他新型的非易失性存储器还存在以下缺点。

- MRAM工作原理依赖磁性，磁性材料在200° C左右的温度环境下会丧失磁性，而在制造和集成工艺的过程中，温度通常会达到400° C；并且MRAM品质不容易控制，如果磁性薄膜系统没有良好的均匀性，会导致写入或读取发生错误。
- FeRAM随工艺缩小的能力比较差，存储密度不够高，在高密度非挥发型存储器领域尚且不能和闪存竞争[17]。
- 虽然RRAM具有可缩小性好、操作电流低、读写速度快、阻态保持特性好等特点[18]，但其仍处于开发的初级阶段，其开关阻变机理不够清晰。

目前研究较成熟的PCM被认为是最有可能取代DRAM的非易失性存储器。PCM以硫系化合物GST材料为存储介质，利用纳米尺寸的相变材料在晶态（材料成低阻状态）与非晶态（高阻状态）时呈现出的阻值差异实现数据存储[19]，通过给上下级加一定的电压，使相变材料在晶态和非晶态之间转变。高阻下非晶态表示二进制“0”，低阻下晶态表示二进制“1”，从而能够写“0”或“1”。综合而言，PCM具有如下特征[15,20]。

（1）高存储密度

NOR flash和NAND flash结构中，门电路厚度固定，需要高于10 V的电源供电，导致其存储器体积很难缩小，而CMOS逻辑门只要1 V或者更少电源即可。根据摩

尔定律，存储器缩小一代，密集程度将提高一倍[21]。PCM可以将不同的电阻区组合在一个存储单元内，存储一个或以上的字节。其存储单元小，相变材料体积小，具有很强的缩放性，从而存储密度提升，内存容量扩大。

（2）非易失性

PCM利用相变材料（如硫系化合物合金材料$Ge_2Sb_2Te_5$）的电阻值来保存数据，不需要像DRAM一样通过电容的充放电来表示数据，也不需要通过周期性的刷新操作来维持存储单元内的数据。掉电后数据存储期限可达10年之久。

（3）按位寻址

PCM具有按位存储的特性，其单元值可直接由“0”变为“1”或由“1”变为“0”，不需要单独的擦除操作，可以降低能耗，节省时间。这与传统的DRAM按字节寻址略有不同，只需更改少许内存管理策略。

（4）低能耗

PCM芯片是由相变材料构成的，漏电能耗极少，几乎可以忽略不计，也不需要DRAM周期性地刷新电流。PCM最大的特点是，在大数据应用的环境中，相对于DRAM，能节约海量的能源消耗。

2.4 PCM的缺陷及大数据应用的需求

虽然PCM是最理想的内存选择之一，但是PCM在读操作和耐写度两方面存在明显缺陷。一是读写不对称。PCM读取时延约为200～300 ns，具有与DRAM相近的读取带宽。但是PCM的写速度较慢，是DRAM的1/10[22]，虽然PCM的写速度比闪存快，但在大数据应用要求低延时的背景下需要考虑如何减少PCM上的写操作以提升系统性能。二是耐写度有限。过多的写操作（10^6～10^8次）会导致PCM器件单元失效。这意味着，在最理想的情况下，一块16 GB的PCM芯片的寿命为10年左右[22]，但由于写操作的速率不同或者写操作的分布不均匀，PCM芯片的寿命会进一步地缩短。

鉴于此，在大数据应用环境下，还需要对PCM进行以下两方面的提升及优化，以更好地满足大数据应用的需求。

（1）减少PCM上的写操作

在实际应用中，从系统的性能考虑，对内存的读写速度有迫切的需求。然而PCM存在读写性能不对称的问题，写请求会导致读时延延长2.3倍[23]。

当有许多写操作发生时，一个较高的写时延能够通过缓冲区和智能调度来解决。但是，当一个写请求被调度到一个PCM块上时，如果这个块在写操作完成之前发出读请求，那么这个读请求就需要等待，因此，写请求会引起读请求时延的延长。和写访问请求不同的是，读访问请求是系统时延的关键，读操作的延缓会对系统的性能造成显著的影响。参考文献[23]中的基准系统的读时延为2 290个周期，是读写竞争较少的系统（1 000个周期）的3倍。如果写时延缩减到1 000个周期，则读时延会缩减到1 159个周期，这表明竞争主要是由写请求引起的，并且是导致读操作效率降低的主要因素[23]。所以，减少发生在PCM上的写操作可以减少操作时延，提升系统性能。

（2）降低PCM的写能耗

PCM不仅存在读写性能不对称的问题，还存在读写能耗不对称的问题。PCM使用的相变材料通过热量的应用来转换存储单元内“0”和“1”的状态。例如相变材料$Ge_2Sb_2Te_5$（GST），当其温度超过其结晶温度（300° C左右）但在其熔化温度（600° C左右）之下时，就会进入结晶状态来表示逻辑上的“1”；当热度超过熔化

温度，GST就会进入非结晶状态来表示逻辑“0”（亦即reset状态）[24]。因此当对一个PCM存储单元进行写操作（set和reset操作）时，在位线①上需要不同的电流和电压，并且需要不同的完成时间。reset操作需要最高水平的电压在短时间内熔融相变材料，使PCM单元变成非结晶的状态。set操作通过长时间的低电压使存储单元结晶化。因此对PCM的读写操作所需要的能耗存在较大的差异，读操作的能耗与DRAM相近，写操作的能耗却比DRAM大[15]。

由于在PCM上进行写操作会消耗大量的能源，如果不降低PCM写能耗，在一定的写次数之后，PCM内的存储单元就会被损坏，从而引发如下两个方面的问题。

- 影响PCM的寿命。应用通常会有对内存系统进行分布不均匀的写操作，这会导致系统的寿命急剧下降，比在理想状态下完全均匀地分布在PCM上的写操作的系统寿命缩短1/20左右。在大数据应用和分析的全球化特征下，计算系统要承受更多的或者分布更不均匀的写操作，如果不对这些写操作进行处理和恰当的部署，那么计算系统就会过早地损毁，造成不可估量的成本开销。
- 影响PCM安全性。PCM某些单元损坏后可能会使原本存储在这些单元的数据丢失。一个恶意的攻击程序可以使PCM内存在其指定的某一行或小范围内的某些行进行反复写操作，致使PCM的存储单元在一分钟内出现故障甚至损坏[25,26]。此外，如果操作系统被攻击者攻破，那么虚拟地址到物理地址的映射就会很容易被识破，攻击者就可以做一个简单的程序，通过将大量数据写到精心挑选的缓存行，使缓存不断刷新其数据。

运用合理的策略优化PCM的缺陷所带来的问题，成为当前研究的热点之一。

① 存储矩阵中的数据输出线

3 PCM/DRAM混合存储架构及优化策略

3.1 PCM/DRAM混合存储架构的优势

PCM存在高存储密度、非易失性、低能耗等优点，而DRAM具有读写操作速度快的优势，两者结合，则可能既规避PCM读写不对称等劣势，又弥补DRAM低存储容量、易丢失的缺陷，从而出现了PCM/DRAM混合存储架构。

PCM/DRAM混合存储架构就是用PCM芯片和DRAM芯片共同构成内存系统，以往对该架构的研究主要分成PCM/DRAM同级混合存储系统和DRAM作为PCM缓存的内存系统。本节主要讨论PCM/DRAM混合存储架构在性能和能耗方面与传统DRAM内存系统相比存在的优势。

（1）性能

大数据应用在传统的DRAM内存架构下运行，由于DRAM通过充放电来表示数据的特性，所以应用需要等待多个周期才能存取数据进行读写操作。而在PCM/DRAM混合存储架构中，影响系统性能的主要因素是发生在PCM上的写操作所带来的时延以及内存发生缺页错误时需要等待的周期时长。IBM公司的研究提出了基于PCM和DRAM的混合存储架构[22]，使用PCM能最大限度地扩充内存的容量，将快速的DRAM放在PCM内存和处理器之间，作为内存中的缓存区，提升系统性能。通过用更大的PCM内存和3%PCM内存块大小的DRAM构成混合存储架构来打破DRAM和PCM在时延方面的差距。研究中还提出了延迟写管理（lazy-write organization）作为混合内存架构的管理机制，通过减少对PCM的写操作来

克服PCM写速度较慢的缺点。同时还采取了行级回写(line-level write back)、细粒度磨损均衡(fine-grained wear-leveling)、页级分流(page level bypass)等机制区分数据块的访问频繁性，减少PCM中的写操作。实验结果表明这些策略能显著减少缺页错误的发生，系统性能与传统存储架构相比提速3倍。

(2)能耗

能耗是计算系统在处理大数据应用时主要考虑的因素之一。DRAM内存系统的漏电效应导致处理大数据时会引起很高的能耗，如何平衡能源效率和系统性能是目前PCM/DRAM混合存储系统的热门研究方向。在计算系统中，图形处理器(graphics processing unit, GPU)用于在通用内存中处理大规模的并行计算，这些并行计算会引起很多写操作，由于PCM的读写不对称，写操作会引起更多的能耗(354%)[27]，Wang B利用PCM/DRAM混合存储架构，通过硬件和编译器优化GPU大规模并行计算环境下的能源效率。该方案通过调整硬件的构成和编译器的功能，利用一个基于并行处理的数据迁移框架完成数据迁移，并利用编译器抉择数据迁移和数据部署的计划，避免引起额外的读写操作，消耗大量能源。最终与只有DRAM构成和只有PCM构成的并行计算内存系统相比，在系统性能损失不到2%的情况下，能源效率分别提升了6%和49%[27]。

3.2 混合存储架构的优化策略

基于以上论证，PCM/DRAM混合存储架构在能耗和性能方面能对大数据应用进行优化，但是PCM读写不对称和耐久度有限的缺陷会使混合存储架构在大数据应用时出现时延高、寿命短等问题，下文将针对能耗和性能这两方面讨论优化策略和算法。

3.2.1 减少PCM上的写操作

由于PCM的读写性能和能耗不对称，过多的PCM写操作会引起额外的系统的内存访问开销和能源开销，减少部署在PCM上的写操作不仅可以延长PCM的寿命，也可以提升PCM内存系统在应用环境下的性能，减少应用中的能耗开销，达到更好的能耗优化效果。目前，减少PCM上的写操作主要分为两个方面：冷热数据划分和读写倾向划分[15]。

(1)冷热数据划分

根据数据被访问的频率和读写操作的次数可以将数据划分为冷数据和热数据，将冷数据存放在PCM上，将热数据存放在DRAM上，就可以将写操作次数更多、访问更频繁的数据从PCM迁移到DRAM上，同时也就将写操作迁移到DRAM上，减少PCM上的写操作。

为了提升PCM系统性能，延长PCM系统的寿命，Lee S等人[28]提出了基于脏数据位和写频繁度的时钟算法 (CLOCK with dirty bits and write frequency, CLOCK-DWF)。算法利用PCM和DRAM同级混合存储架构，将读请求和干净页面(clean page)部署在PCM上，将写请求和脏页面部署在DRAM上。当一个写操作发生在PCM时，这个页面就会被标记为脏页面，CLOCK-DWF算法就将这个页面从PCM迁移到DRAM上。如果此时DRAM为满，那么就会选中一个页面迁移到PCM上。算法通过统计页面的写频繁度区分页面是热还是冷，并通过脏数据位统计页面的写频繁度。如果一个候选页面的脏数据位为“1”，那么算法就会将这个页面标记为干净页面但在写频繁度上加1，如果候选页面的脏数据位为“0”，那么算法就会检查这个页面是热还是冷，如果为冷就迁移到PCM上。实验结果表明，与

DRAM作为PCM缓存的混合存储架构相比，算法平均能减少35.4%的写操作，与传统的CLOCK算法相比，平均减少14%的写操作。

为了避免冷热数据划分带来的迁移导致过多的写操作，Lee M等人[29]提出了迁移优化的页面替换时钟（migration-optimized CLOCK，M-CLOCK）算法。为了有效地区分DRAM上写频繁页面和读频繁页面，M-CLOCK算法利用了两个时钟指针：D指针和C指针。D指针负责管理热脏页面，C指针负责管理短期内不具有写倾向的页面。当一个候选页面被写操作重新访问时，M-CLOCK算法就会通过写倾向位和脏数据位确定这个页面是否为热脏页面。当DRAM为满，D指针就会在脏热页面里选择一个具有最低写频繁度的页面，如果页面写倾向位为“1”，那么这个页面就会被选为候选页面，否则就将页面写倾向位设置为“0”，然后指向下一个页面。如果D指针找不到最低写频繁度的页面（写倾向位为“0”），那么就会由C指针在干净页面内选择候选页面。M-CLOCK通过这样的迁移方法，将DRAM中的读冷页面迁移到PCM上，与过往的算法相比，能减少最多98%的写操作，最高提升34%的系统访问时间。

为了在尽量小的性能代价下，优化混合存储系统在应用环境下的能耗，避免不必要的DRAM到PCM页面迁移所引起的写操作，Shin D J等人[30]提出了自适应的页面组管理（adaptive page grouping，APG）。算法认为物理距离相近的页面具有相似的访问请求次数，于是根据页面的物理距离，统计页表中各个页面中读写请求的次数，从而决定访问热度，将访问热度相近的页面聚类在一个分组，如果组的平均热度超过热度阈值，则将组设定为热组；如果低于冷度阈值，就设定为冷组；热度阈值和冷度阈值之间的页面为暖组（warm group）。算法将热组部署在DRAM上，冷组部署在PCM上，暖组不发生迁移操作。实验结果表明，算法能有效地减少PCM写操作，且与DRAM系统相比，减少36%的能源消耗，与低时延的PRAM相比，内存访问效率增加了80%。

CLOCK-DWF算法通过数据被访问的频繁度与写访问请求的历史记录，准确地估计数据未来的冷热度，并利用DRAM吸收更多的写操作，但当写请求访问PCM上的一个页面时，就会将这个页面迁移到DRAM上，如果此时DRAM已满，则需从DRAM里面选取冷页面与PCM交换，如果后面这个页面被写请求访问，那么又要将页面换回到DRAM，引起大量的额外读写操作。而且当发生缺页错误时，CLOCK-DWF将大部分的错误页面部署到了PCM上，这也会引起额外的PCM的写操作。

M-CLOCK算法根据数据的访问热度和是否为脏数据来决定数据是否需要迁移，还利用了一个懒惰迁移（lazy migration）来延缓PCM到DRAM的页面迁移，解决CLOCK-DWF的迁移颠簸问题。

APG算法通过将热度相近的页面聚类在一起，并设置阈值，将页面组划分为冷热组，但APG不能区分空间相邻的页面的访问频繁度，可能会造成冷页面的迁移，导致额外的内存访问时延。

（2）读写倾向划分

读写倾向的划分主要是根据数据的写操作次数将数据划分为具有读倾向还是具有写倾向，将具有读倾向的数据存放在PCM上，将具有写倾向的数据存放在DRAM上，可以减少部署在PCM上引起写操作的数据，达到减少PCM上写操作的目的。

以往的研究发现，大多数高速缓存中的未命中是由于被访问的数据块同时被

重叠地映射到相同的缓存组，而在DRAM作为PCM的缓存的混合存储架构中，缓存的未命中会导致数据写回到PCM上。基于此，Khouzani H A等人[31]提出了基于冲突的页面分配算法(conflict-aware proactive page allocation algorithm)，利用虚拟页面映射到物理页的灵活性，根据页面分段信息和DRAM中的未命中，通过设置不同的DRAM组，重新分布DRAM中有很高写倾向的页面。该算法由两个部分组成。首先，内存控制器（MC）负责记录DRAM上未命中的页面冲突。其次，当内存发生页面错误时，操作系统负责比较DRAM中的内存块并选出具有更少页面冲突的内存块。这个部分能决定硬件的构成并展示在硬件成本和算法复杂度方面如何最大限度地发挥算法的效率。当一个访问请求在DRAM上未能命中，而在PCM上命中了，这意味着，DRAM的大小或者相关性提高，就可以避免这次未命中并将其记作一个未命中的冲突。该算法为了记录DRAM内存块上的这些不同数量的未命中的冲突，在每个内存块上设置了一个基于硬件的计数器。由于这些冲突数量可能会非常大，基于存储开销和访问时间的考虑，该算法只在DRAM内存块添加了一个2位饱和计数器用于区分更高冲突性和更低冲突性的内存块。内存管理器负责管理DRAM和PCM之间的通信，也负责管理内存块中的计数器。当发生更高优先级冲突时（即DRAM未命中而引起的写回），计数器的值就增加2；如果是常规冲突，计数器的值增加1。当发生页面错误时，如果请求的页面不属于文本段，则将分配给该页一个更低冲突性的内存块。由于该算法只使用了一个低开销的2位饱和计数器，直接比较所有计数器来找出最小冲突块的代价是昂贵和不必要的，因此该算法还利用了经典的时钟算法来找出最低冲突性的内存块。当发生缺页错误时，时钟指针以循环的方式逐个扫描DRAM内存块中的计数器。如果计数器的值不为零，则计数器的值减1；当时钟指针指向计数器值为零的内存块时，这个内存块就会被选中，并作为候选的具有更低冲突性的内存块，保存发生错误的页面。在记录驱动的实验中证实了该算法能有效地减少PCM上的写操作（25%）和提高DRAM中的命中率（减少27%的未命中），因此，同时也能提高DRAM/PCM混合内存的性能和寿命。

为了减少PCM上的写操作并且保持稳定的系统应用性能，Wu Z L等人[32]提出了基于访问形式预测的LRU（access-pattern-prediction-based LRU，APP-LRU）算法②。APP-LRU算法中包含3个链表：LRU链表、PCM链表和DRAM链表。LRU链表用于管理PCM和DRAM，当一个页面被访问时，就会将其放置在LRU链表中最近最频繁使用的一段。PCM链表和DRAM链表中所有的页面都会被分成若干含有一定数量页面的组。PCM链表中的同一组内所有的页面都具有相等的写次数，而DRAM链表中的同一组内页面具有相等的读次数。PCM链表（DRAM链表）头部的组的页面具有最大的写（读）操作次数，当一个页面从磁盘里读出来或从DRAM迁移到PCM（或从PCM迁移到DRAM）时，这个页面就会被放置到链表的尾部。当读取或更新DRAM中的页面时，该页面将会从属于PCM链表中的组迁移到属于DRAM链中的组。APP-LRU就是通过这样的迁移方法来减少PCM上的写操作的，在有效地减少PCM上写操作的前提下，与CLOCK-DWF算法和LRU算法相比，其迁移操作是传统算法的1/6左右。

为了在减少PCM上的写操作的同时保证系统的命中率，确保系统在应用中的性能，Chen K等人[33]提出了保持命中率的

② LRU是least recently used的缩写，即最近最少使用。LRU算法是内存管理的一种页面置换算法

LRU (maintain-hit-ratio LRU, MHR-LRU) 替换算法。算法使用LRU链表管理混合内存架构中的内存页面，所有页面根据其最近的使用时间排列在内存中。当出现缺页错误时，在最近最少使用位置的页面就会被选中。在DRAM里，算法使用了一个基于DRAM写数据的LRU链表（DRAM write-aware LRU list，DWL），DRAM上的页面根据最近的写倾向时间排列在这个链表中。当发生页面错误并选中一个候选页面时，MHR-LRU就会检测页面的访问是读还是写，并找出候选页面的位置，如果页面的访问模式为写并且部署在PCM上，那么算法实行页面的迁移，释放PCM中的候选页面，并且将DWL链表中在最近最少使用位置的页面迁移到PCM上，那么这个提出写访问请求的页面就会被部署到DRAM上。实验结果表明，与其他算法相比，MHR-LRU算法在保证命中率的情况下平均减少6.48%的PCM上的写操作。

将具有写倾向的页面从PCM上迁移到DRAM上可以有效地减少PCM，基于冲突的页面分配算法利用DRAM作为缓存，吸收了具有写倾向的页面，减少了PCM上的写操作。在此前提下，还减少了由于缓存的未命中引起的对PCM的写操作，最终延长了系统的寿命，同时也提升了系统的性能。但是对于利用DRAM作为PCM的缓存的混合内存架构，由于DRAM只作为系统的缓存，在后续的应用运行中，如果PCM上的页面发生读写倾向的改变，那么就无法将PCM上的页面迁移到DRAM上。

APP-LRU和MHR-LRU算法都利用了PCM/DRAM同级混合内存架构，APP-LRU算法通过元数据表记录页面的访问历史区分页面的读写倾向，MHR-LRU算法通过LRU链表管理读写倾向划分后按照使用时间排序的页面。APP-LRU由于没有考虑页面的使用频繁度，所以可能会迁移最近使用比较少的页面，降低系统的命中率，增加额外的时间开销；MHR-LRU算法通过使用频繁度的排序，保证了系统的命中率，但是仅能将DRAM上读倾向使用不频繁的页面迁移到PCM上，不能将PCM上写倾向使用频繁的页面迁移到DRAM上，不能更有效地减少PCM上的写操作。

3.2.2 磨损均衡

在很多大数据应用场景中，写操作会集中在内存的某一行或某一页，承受过多写操作的区域由于耐久度有限会更早地损坏，从而缩短PCM的整体寿命。磨损均衡（wear leveling）就是研究如何使PCM上的写操作均匀分布，以延长计算系统寿命的算法。现有的具有代表性的工作有以下几个。

（1）Start-gap磨损均衡

由于现有的磨损均衡算法需要一个很大的表来追踪发生在PCM上写操作的次数，Qureshi M K等人[25]提出了Start-gap磨损均衡算法，利用一个简单的物理机构，既避免了已有的磨损均衡算法所需的存储空间和操作时延，同时也尽可能地达到了理想磨损均衡算法下PCM的寿命。算法利用了两个物理寄存器Start和gap，还用了一个空隔行（gap line），每发生100次写操作就移动一次gap指针和空隔行，同时gap寄存器中的数值减1（gap寄存器初始值为当前总行数减1），每次写操作都从Start指针开始，其基本过程如**图1**所示。当gap指针指向0时，Start指针和寄存器加1。Start-gap磨损均衡算法就是通过在连续不断的PCM空间上不停地移动空隔行（不进行写操作）达到磨损均衡效果的。但由于写操作通常聚集在相邻的行，实验过程中，只能达到理想情况53%的效果，所以需要随机地分配地址，将写操作均匀地分布

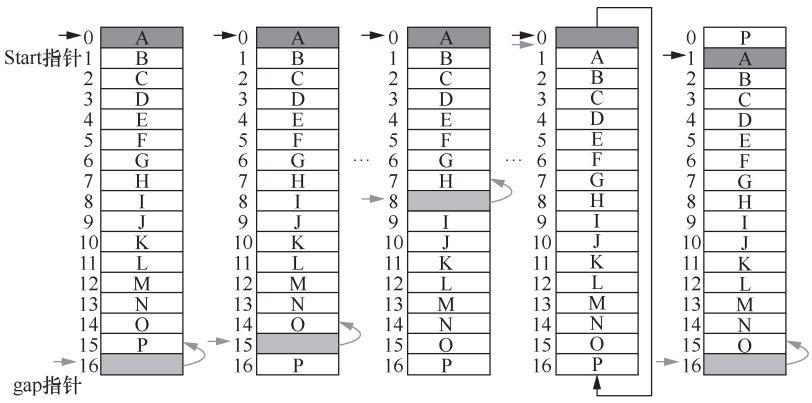


图1 Start-gap 磨损均衡算法

在不同的区域。Start-gap磨损均衡算法在此基础上还分别引入了密码学技术Feistel网络和一个随机可逆的二维矩阵，达到逻辑地址对物理地址的随机代数式映射。最终，在采取了随机地址映射的方法后，Start-gap磨损均衡算法能达到超过97%的理想状况下PCM寿命的效果。

(2)软件实现的磨损均衡算法

软件实现的磨损均衡算法主要考虑的问题是，通过程序或者编译器均匀分配PCM上的写操作，并考虑内存访问的时间。Hu J等人[34]提出的基于嵌入式系统的软件实现磨损均衡，在数据已经被部署在混合存储系统中，并且已经知道每个数据将会引起多少写操作的前提下，利用程序实现均匀分配PCM上的写操作。循环的开始和结束将程序划分为不同的区域。首先用最优数据分配(optimal data allocation, ODA)算法③配置数据的分布，以获得分配在PCM上的变量。然后利用一个数组W记录PCM上每个地址的写操作次数。然后软件实现的磨损均衡算法将PCM按照地址的前后分为两个组。对于第一个组的数据D_j，程序先通过数组W获得地址$addr_j$的写次数，然后对地址$addr_j$的写次数和数据D_j的写次数进行求和，再与阈值进行比较，如果写次数的和小于或等于阈值，那么数据D_j将会留在原地址；否则，就从W里面找出写次数最少的地址，并计算该地址的写次数与数据D_j写次数的和值，如果未超过阈值，则将数据D_j迁移到这个地址上，若超过阈值，则要重新设置阈值。对于第二组的数据D_k，程序直接从W找出写次数最少的地址，并计算该地址的写次数与数据D_k写次数的和，如果未超过阈值，则将数据D_j迁移到这个地址上，若超过阈值，则要重新设置阈值。最后实验中，软件实现的磨损均衡算法在先利用ODA算法进行数据部署、在可接受的额外的应用时间开销(5.46%)的前提下，使PCM的寿命时间平均延长了3.13倍。

(3)自适应磨损均衡算法

为了减少和均匀分配PCM上的写操作，Park S K等人[35]提出了自适应磨损均衡算法(adaptive wear-leveling algorithm)。由于脏数据的清除会引起PCM上的写回(write back)，增加应用过程PCM上的写操作，因此首先将DRAM(3%)作为PCM的缓存并将DRAM缓存分为两层，分别处理脏数据(dirty data)与

③ 该算法无差别对待PCM的地址，总是将PCM上第一个可用的空间分配给变量

干净数据(clean data)。第一层利用传统的LRU算法决定脏数据和干净数据的替换顺序,第二层通过统计脏数据的写操作次数,将写操作次数最少的脏数据替换出去。基于操作时间与一次所要交换的页面数量的考虑,自适应磨损均衡算法第二步提供了一个自适应的多数据交换和移动的框架以实现磨损均衡。通过周期性地检测最大写入数的增量,查询写入访问是否倾向某一行或某一页,以此动态地调整页面交换的负载模式。自适应磨损均衡算法的最后,在页面或行交换的时候,将页面或行内所包含的脏数据也先替换出缓存,并写回到PCM上,避免了页面或行交换和脏数据的重复写操作。通过这3个步骤,自适应磨损均衡算法能将以往的磨损均衡算法下的PCM寿命从0.68年提升到5.32年。

(4)基于行的映射和循环利用磨损算法

已有的错误修改指针(error correcting pointers, ECP)算法在出现了不可修复的错误行的时候就将其标记为不可用,这样就会造成PCM上的空间不连续,不能与Start-gap磨损均衡算法组合起来,Jiang L[36]提出了基于行的映射和循环利用(line-level mapping and salvaging, LLS)磨损均衡算法。首先将PCM上一定的空间分为28个数据块作为主空间,其余的作为备用的PCM空间。用已有的循环利用算法ECP对出现错误的行进行修复,当出现第一个ECP无法修复的错误行时,就启用LLS,LLS将主空间的错误行部署到备用的PCM的空间,这样错误行就能被标记并且重映射到一个健康的行中。当PCM主空间里损坏的行比备用PCM空间里还未损坏的行多时,就会激活PCM的大小调整,以提供PCM上连续的地址空间。LLS磨损均衡算法比ECP算法下PCM寿命平均延长了24%。

(5)基于桶和基于数组的磨损均衡算法

基于时间和空间复杂度的考虑,Chen C H等人[37]提出了基于桶和基于数组的磨损均衡算法。基于桶的磨损均衡算法如**图2**所示。先将PCM内的页面以桶的形式按照页面的磨损程度聚类,在桶内的页面按照写次数的多少排列,再将这些桶分为两个链表,即空闲链表和正在被占用的链表。当需要调用页面时,先调用空闲链表中磨损程度最低的页面,如这个桶为空,则将正在被占用链表的基桶(磨损程度最低)页面的数据写入写次数最多的空闲页

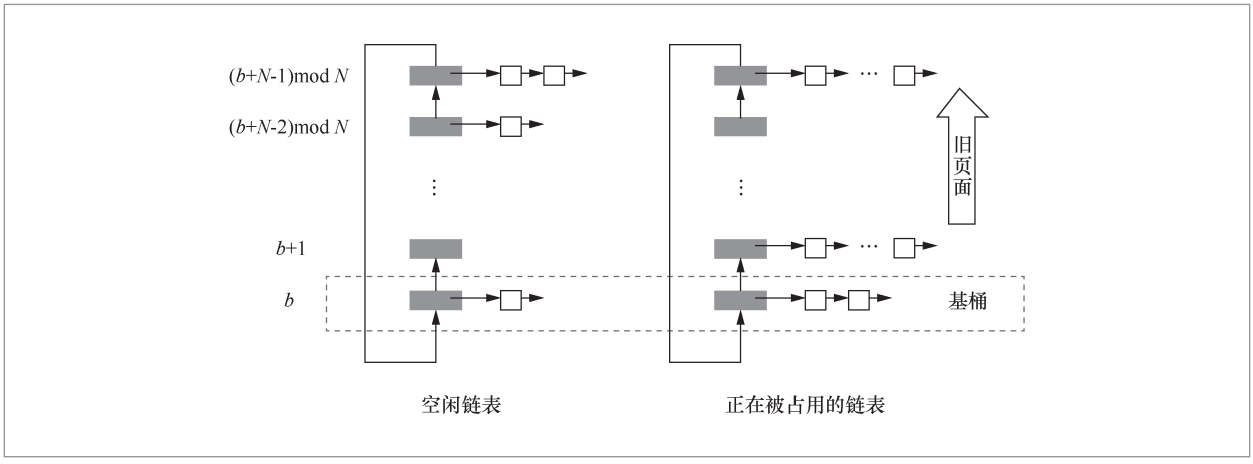


图 2 基于桶的磨损均衡算法

面，然后再调用这个页面。基于数组的磨损均衡算法如图3所示。在物理页面上添加了两个物理计数器b、m和一个指针，当页面的磨损程度超过了初始设定的阈值，就以指针为中心，将邻近的若干个页面磨损程度最低的页面交换。最终，基于桶和基于数组的磨损均衡算法下的PCM内存寿命达到了完全磨损均衡理想状态下的80%。

磨损均衡算法小结如下。

- Start-gap算法将PCM内存行移动到邻近的地址空间，利用两种随机映射地址的技术，分配区域集中的写操作。
- 软件实现的磨损均衡算法通过数组统计数据的写次数，利用软件对数据进行迁移，但由于该算法要先利用ODA算法对数据进行分配，所以会引起额外的内存访问时间开销。
- 自适应的磨损均衡算法能够根据写操作的分布情况，自动调整数据的交换和迁移，在尽量避免不必要的交换和迁移引起的额外的写操作前提下，达到了良好的延长PCM寿命的效果。但是该算法需要大量的附加硬件支持，每512 MB的PRAM就需要32 MB的内存空间以支持算法。
- LLS算法通过对硬件故障修复技术ECP的改进，使PCM上的可使用地址恢复连续性，得以和Start-gap算法结合，同时实现PCM的回收利用和磨损均衡的效果。
- 基于桶和基于数组的磨损均衡算法仅能达到理想状态下PCM寿命的80%，会引起2%的额外写操作，但由于算法不需要跟踪内存页面的写频繁度，所以对系统的性能影响几乎为零。

3.2.3 PCM的故障处理

传统的DRAM技术拥有容错技术，在没有寿命限制的情况下，内存系统利用错误检查和纠正（error checking and correcting，ECC）编码修复DRAM中的瞬态故障。然而在PCM内存系统中，由于在一定次数的写操作之后，存储单元很快被磨损，基于电阻内存的ECC修复编码快速失效。一旦存储单元出现了故障，该单元上的数据就无法继续使用，PCM也会被认定为已经损坏，所以，在PCM/DRAM混合存储系统中，需要新的容错机制以修复内存存储单元的故障。为了保证存储在PCM上的数据的安全性，修复PCM上的故障存储单元，目前主要的工作如下。

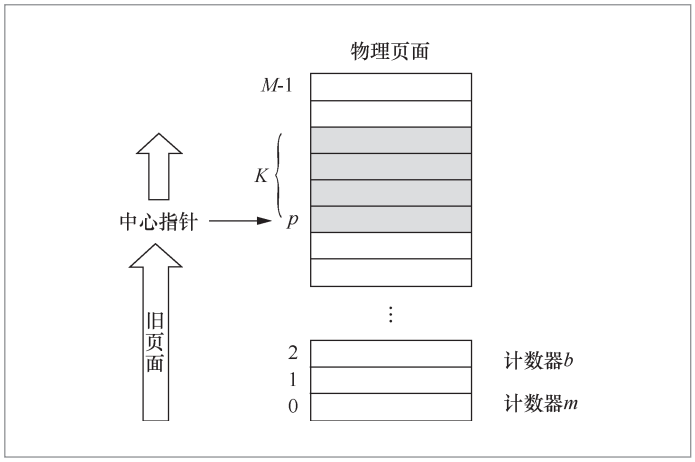


图 3 基于数组的磨损均衡算法

（1）ECP算法

为了尽量减少写磨损，处理永久性的存储单元故障，提高内存系统的寿命，修正早期的存储单元故障，Schechter S等人[38]提出了ECP算法。ECP算法通过将故障单元的位置永久编码到表中并分配单元替它们来纠正错误。图4（a）显示了ECP算法一个最简单的应用，即对一个位的修正，当一个位出现故障时，这个位就会被标记为满，修正指针就会指向这个位，然后就会用一个新的存储单元存储这个位原来具有的值。当需要对内存单元进行n位修正时，如图4（b）所示，当出现第一个故障时，就利用修正记录位0进行修正。当替代的单元也出现了故障时就用图4（c）的方法进行修

正，当修正指针发生故障时就会利用图4（d）的方法进行修正。ECP算法在大量的写操作发生后，仍能保持页面的健康性。

（2）动态复制的内存

为了在硬件和操作系统层面有效地延续PCM的物理可使用性，Ipek E等人[11]提出了动态复制的内存（dynamically replicated memory，DRM）算法修复PCM上的硬件故障。为了便于动态复制，DRM引入了一个新的间接寻址层，位于系统的物理地址空间的PCM和真实地址空间的PCM之间。物理地址中的每一页都被映射到一个未使用过的无故障的真实页面，或两个有故障的但在同一个位上没有故障的兼容性页面，因此，可以配对在每一个位上进行读写操作的物理页。为了完成这样的映射，DRM算法在PCM上存储了3个独立的副本表，若对第一个副本表的复制出现错误，那么系统就会尝试第二个副本表、第三个副本表；如果所有的副本表都出现故障，相应的物理页面都会被弃用。为了从映射之后的PCM地址获得数据，DRM算法利用硬件追踪真实地址，并利用操作系统保证没有不兼容的页面被配对在一起。DRM算法与传统的内存错误修复机制相比，在过程变化程度（process variation，以下用CoV表示）不同的情况下，能不同程度地延长PCM的寿命。当CoV=0.1、CoV=0.2和CoV=0.3时，DRM能分别将PCM的寿命延长至1.25倍、2.7倍和40倍。

PCM的故障处理技术算法小结如下。

- ECP算法利用操作系统对出现故障的单元进行追踪，将其标记为不可用并取回数据。ECP算法标记所有的出错单元，因此当故障单元数超出了一定的限制后，ECP算法便无法进行修复，并且会使PCM上的可使用地址空间不再连续，无法继续使用其他利用随机地址映射方式的优化技术。
- DRM算法动态地配对两个故障页面，从中获得一个可使用的页面来回复

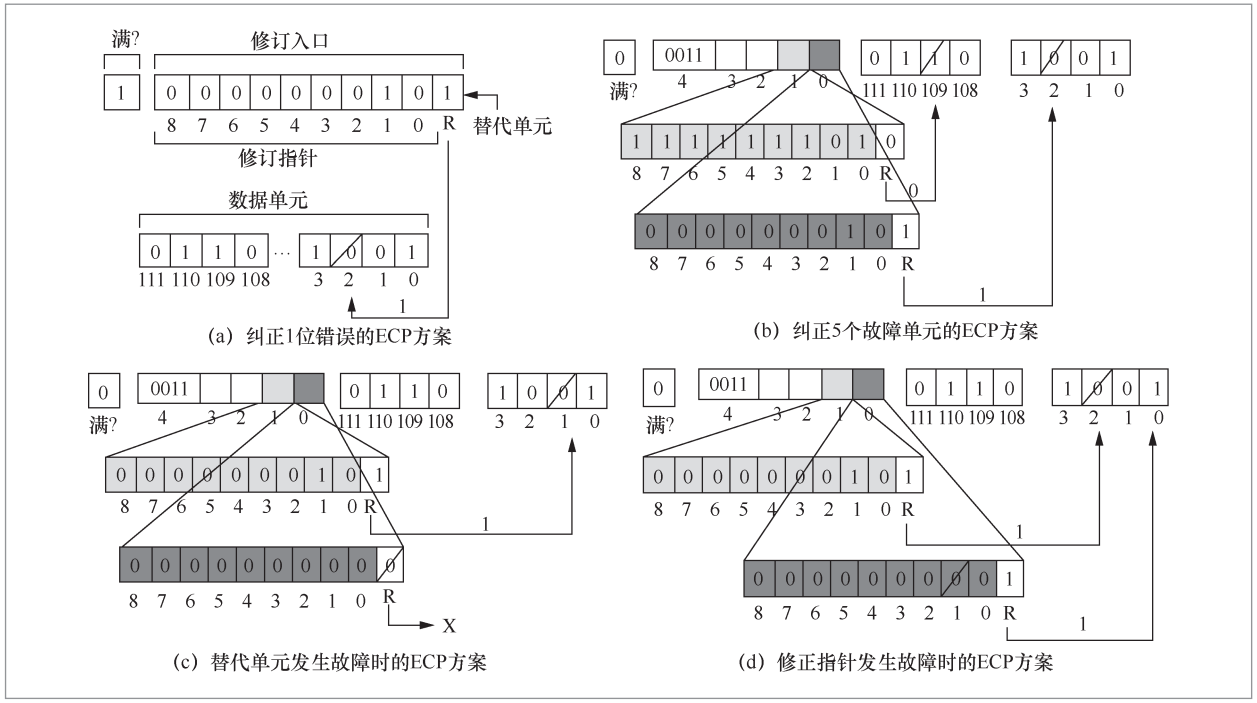


图 4 ECP 的应用例子

PCM早期的故障页面，但动态配对需要检测两个故障页面的故障单元数和故障单元的位置，如果故障页面的故障单元过多，会导致大量额外的时间开销和能耗开销。

所以故障修复处理技术必须与现有的磨损均衡算法和减少PCM上写操作的算法结合起来，才能更好地延长PCM的寿命，并节省时间与能源开销。

4 混合内存未来的优化及研究方向

4.1 已有工作的不足

现有工作存在以下不足。

● DRAM作为PCM的缓存的混合内存架构，如果没有额外的存储容量的支持与开销，系统的性能优化效果不能得到保证，与存储容量较小的DRAM系统相比，性能的优化效果显著，但与存储容量相近的DRAM系统相比，性能上还是存在微弱的劣势。

● 在PCM和DRAM同级混合内存架构中，数据部署在不同的存储介质中，而由于PCM和DRAM的性能属性不同，对数据进行冷热度和读写倾向划分时，需要统计数据的读写历史次数，这会导致额外的时间开销和能耗开销，而且数据的划分会导致迁移操作，这些迁移操作会引起额外的读写操作，降低PCM和DRAM同级混合内存架构的优化效果。

● 对于读写倾向的划分和冷热数据划分，冷页面的迁移会带来PCM上不必要的写操作，同时PCM与DRAM之间的页面迁移次数与迁移时机的不恰当也会导致迁移颠簸。

● 在以往磨损均衡的研究工作中，数据的交换和迁移会导致额外的PCM上的写操作，目前可依靠良好的阈值设定来控制数据的交换、迁移的次数和频繁度，但静态的阈值设定无法满足应用写操作的动态变化，动态的阈值设定则需要硬件、操作系统或者软件来支撑，带来额外的成本开销与时间开销。

● 在PCM的故障处理中，最显著的问题是，当PCM的故障单元超过一定的限制后，这些技术就无法再使用，而将磨损均衡与故障处理的技术结合起来需要提供连续的PCM可使用的地址空间。

4.2 PCM/DRAM混合内存的优化研究方向

在大数据应用优化的需求背景下，对PCM/DRAM混合内存管理策略的研究，关键在于结合应用特征以及算法策略，并进行适当的调整以适应不同的应用指标和架构环境。例如，磨损均衡中所提到的LLS算法，就是通过改变故障修复策略，在PCM上保持了连续的空间，满足了磨损均衡算法的需求。基于PCM和DRAM的关系，可以在两种不同的架构方式下考虑。

在DRAM作为PCM缓存的混合内存架构下，由于PCM的空间地址是独立的，所以可利用PCM的故障修复处理技术，延长系统的寿命。DRAM作为PCM缓存的混合内存架构，可采用磨损均衡算法对PCM内存上的写操作进行分布，延长系统的寿命。此外还可充分发挥DRAM缓存能吸收PCM上的写操作的优势，允许磨损均衡的迁移和交换操作引起的额外写操作。同时扩大内存PCM的容量，使系统的性能得到提升，也可以承担由于控制磨损均衡所需要的硬件开销。在保证了系统的寿命的前提下，利用DRAM作为PCM缓存的混合内存架构自身的性能优势，对大数据应用进行优化。

在PCM/DRAM同级混合内存架构下，由于PCM动态（读写操作）能耗比DRAM高，PCM的写操作效率是DRAM的1/8，所以在PCM/DRAM同级混合内存架构中，需要将写倾向的页面或其他存储结构维持在DRAM上，并且要维持系统的命中率，避免迁移之后的缺页错误。利用DRAM动态能耗低和读写操作性能优化的特性，避免DRAM的漏电。对于在PCM/DRAM同级混合内存架构上减少PCM写操作的算法，可以动态地统计DRAM上和PCM上页面的写操作，根据这些页面的冷热度聚类在一起，再依据它们的写操作次数按升序排列在某一数据结构中，当PCM发生写操作时，释放PCM被选中页面，从DRAM中将写次数最少、最冷（最空闲）的页面迁移到PCM上，若DRAM无空闲页面，就将PCM中写次数最多、最热的页面与DRAM中写次数最少、最冷（最空闲）的页面交换。

基于上述分析，可以将未来的研究方向[39]概括为以下几个方面 。

● 研究内存容量配置对性能的影响。混合内存可以按比例为1∶1的DRAM和PCM容量比配置，但PCM具有良好的扩展性，因此未来的内存容量配置也可能为较大容量PCM和较小容量DRAM的混合，所以需要研究评测内存容量配置对内存性能的影响。此外，结合大数据应用的具体特征，例如数据的读写频率与模式等，获得最佳内存配置在性能与成本间的平衡。

● 研究不同结构的混合内存DRAM/PCM。混合内存分为平行结构和层次结构。平行结构可避免存储相同的数据，更好利用DRAM的容量；层次结构是DRAM作为PCM的缓存，能更好地缓存频繁访问的数据，减少DRAM和PCM间的数据移动开销。此外，结合大数据应用的具体特征，探索不同结构的适用性问题。

● 研究PCM的耐写性问题[40]。过多的写操作会导致存储单元磨损，减少PCM寿命，可分析在不同应用的读写、访问、存储的行为下PCM使用的寿命情况，从而从体系结构方面更好地提高寿命及其耐写度。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。

● 研究混合内存能耗问题。PCM静态功耗低于DRAM，但是动态写能耗高于DRAM，因此需研究不同大数据应用下两者内存的能耗差异，并基于这种差异探索应用感知的优化策略，降低内存器件的能耗。此外，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PCM的安全性问题。利用PCM的非易失性，可以在内存中直接对部分数据做持久化。另外，PCM非易失性的特征带来了数据信息泄露的安全隐患。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，然后指导其他数据被正确分配、访问，维护数据的一致性。目前，安全性研究展望主要包括4个方面[41]：融合权限和保护机制、加强程序安全、使用非易失缓存和提供硬件支持保持数据一致性、减少PCM的保持时间以降低数据被窃取的风险。

● 研究数据标签化及数据部署问题。与传统内存系统不同的是，混合存储架构下每种内存介质的性能差异较大，导致在内存页面管理上必须将不同存储介质的页面区别对待，因此合适的数据分类非常重要[15]。需要研究数据标签化（即对数据进行划分）以及数据在混合存储器件上的部署问题。

5 结束语

本文通过分析大数据的应用特征和以PCM为代表的新型非易失性存储器的特点，阐述了混合存储架构在性能和能耗方面与传统内存架构相比存在的优势，并讨论了混合架构的优化算法，总结了未来的优化和研究方向。在未来的发展中，PCM/DRAM混合内存架构可以逐渐取代原有的DRAM架构，成为计算系统的内存。通过扩充内存的容量，提升计算系统在大数据应用背景下的性能，节省计算系统在大数据应用背景下的能耗，满足大数据应用的数字化、全球化、超海量、实时性和中心化等特性。

参考文献:

[1] 马建光, 姜巍. 大数据的概念、特征及其应用[J]. 国防科技, 2013, 34(2): 10-17.
MA J G, JIANG W. The concept, characteristics and application of big data[J]. National Defense Science & Technology, 2013, 34(2): 10-17.

[2] 张引, 陈敏, 廖小飞. 大数据应用的现状与展望[J]. 计算机研究与发展, 2013, 50(S2): 216-233.
ZHANG Y, CHEN M, LIAO X F. Big data applications: a survey[J]. Journal of Computer Research and Development, 2013, 50(S2): 216-233.

[3] 何立民. 大数据时代与嵌入式系统[J]. 单片机与嵌入式系统应用, 2014, 14(1): 1-3.
HE L M. Big data era and embedded systems[J]. Microcontrollers & Embedded Systems, 2014, 14(1): 1-3.

[4] 李涛, 曾春秋, 周武柏, 等. 大数据时代的数据挖掘——从应用的角度看大数据挖掘[J]. 大数据, 2015, 1(4): 1-24.
LI T, ZENG C Q, ZHOU W B, et al. Data mining in the era of big data:from the application perspective[J]. Big Data Research, 2015, 1(4): 1-24.

[5] 孙大为. 大数据流式计算：应用特征和技术挑战[J]. 大数据, 2015, 1(3): 99-105.
SUN D W. Big data stream computing: features and challenges[J]. Big Data Research, 2015, 1(3): 99-105.

[6] 伏琰. 云计算环境下数字资源整合模式研究[J]. 河南图书馆学刊, 2014(11): 126-129.
FU Y. Research on the integration model of digital resources in the cloud computing environment[J]. The Library Journal of Henan, 2014(11): 126-129.

[7] BARROSO L A, HÖLZLE U. The case for energy-proportional computing[J]. Computer, 2007, 40(12): 33-37.

[8] BARROSO L A, HÖLZLE U. The datacenter as a computer: an introduction to the design of warehouse-scale machines[J]. Synthesis Lectures on Computer Architecture, 2009, 8(3): 154.

[9] MEHRZAD. Sponge: portable stream programming on graphics engines[J]. ACM Sigplan Notices, 2011, 47(4): 381-392.

[10] NAWATHE U G, HASSAN M, WARRINER L, et al. An 8-Core 64-Thread 64b Power-Efficient SPARC SoC[C]// IEEE International Solid-state Circuits Conference, December 20-22, 2007, Tainan, China. Piscataway: IEEE Press, 2007: 108-590.

[11] IPEK E, CONDIT J, NIGHTINGALE E B, et al. Dynamically replicated memory: building reliable systems from nanoscale resistive memories[J]. ACM Sigplan Notices, 2010, 38(1): 3-14.

[12] 王丽芳, 齐勇, 蒋泽军, 等. 面向大数据的绿色IT框架能效分类机制[J]. 工程研究：跨学科视野中的工程, 2014(3): 224-232.
WANG L F, QI Y, JIANG Z J, et al. Energy-efficiency classifying mechanism for green IT framework of big data[J]. Journal of Engineering Studies, 2014(3):

224-232.

[13] 慎涵. 一种基于PCM的文件系统设计与实现[D]. 武汉: 华中科技大学, 2013.
SHEN H. Design and implementation of a PCM file system[D]. Wuhan: Huazhong University of Science & Technology, 2013.

[14] 陈纯. 流式大数据实时处理技术、平台及应用[J]. 大数据, 2017, 3(4): 1-8.
CHEN C. Real-time processing technology,platform and application of streaming big data[J]. Big Data Research, 2017, 3(4): 1-8.

[15] 吴章玲, 金培权, 岳丽华, 等. 基于PCM的大数据存储与管理研究综述[J]. 计算机研究与发展, 2015, 52(2): 343-361.
WU Z L, JIN P Q, YUE L H, et al. A survey on PCM-based big data storage and management[J]. Journal of Computer Research and Development, 2015, 52(2): 343-361.

[16] 陆游游, 舒继武. 闪存存储系统综述[J]. 计算机研究与发展, 2013, 50(1): 49-59.
LU Y Y, SHU J W. Survey on flash-based storage systems[J]. Journal of Computer Research and Development, 2013, 50(1): 49-59.

[17] 郭姣姣. 基于氧化铪的阻变存储器性能及机理的研究[D]. 上海: 复旦大学, 2012.
GUO J J. Study on performance and mechanism of resistive memory based on hafnium oxide[D]. Shanghai: Fudan University, 2012.

[18] 万海军. 电阻存储器RRAM的可靠性研究[D]. 上海: 复旦大学, 2010.
WAN H J. Research on reliability of RRAM[D]. Shanghai: Fudan University, 2010.

[19] 沈菊, 宋志棠. 相变存储器驱动电路的设计与实现[J]. 半导体技术, 2008, 33(5): 431-434.
SHEN J, SONG Z T. Design and realization of driving circuit for phase-change RAM chip[J]. Semiconductor Technology, 2008, 33(5): 431-434.

[20] International Technology Roadmap for Semiconductors(ITRS)[R]. 2012.

[21] 章征海. 相变混合存储器的研究与设计[D]. 武汉: 华中科技大学, 2012.
ZHANG Z H. Research and design of PCRAM-based hybrid storage system[D]. Wuhan: Huazhong University of Science & Technology, 2012.

[22] QURESHI M K, SRINIVASAN V, RIVERS J A. Scalable high performance main memory system using phase-change memory technology[J]. ACM Sigarch Computer Architecture News, 2009, 37(3): 24-33.

[23] QURESHI M K, FRANCESCHINI M M, LASTRAS-MONTANO L A. Improving read performance of phase change memories via write cancellation and write pausing[C]// IEEE International Symposium on High Performance Computer Architecture, Janaury 9-14, 2010, Bangalore, India. Piscataway: IEEE Press, 2010: 1-11.

[24] ZHOU P, ZHAO B, YANG J, et al. A durable and energy efficient main memory using phase change memory technology[J]. ACM Sigarch Computer Architecture News, 2009, 37(3): 14-23.

[25] QURESHI M K, KARIDIS J, FRANCESCHINI M, et al. Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling[C]// The 42nd Annual IEEE/ACM International Symposium on Microarchitecture, Dec 12-16, 2009, New York, USA. New York: ACM Press, 2009: 14-23.

[26] WU G, GAO J, ZHANG H, et al. Improving PCM endurance with randomized address remapping in hybrid memory system[C]// IEEE International Conference on Cluster Computing, September 26-30, 2011, Austin, USA. Piscataway: IEEE Press, 2011: 503-507.

[27] WANG B, WU B, LI D, et al. Exploring hybrid memory for GPU energy efficiency through software-hardware

co-design[C]// International Conference on Parallel Architectures & Compilation Techniques, October 7, 2013, Edinburgh, UK. Piscataway: IEEE Press, 2013: 93-102.

[28] LEE S, BAHN H, NOH S H. Characterizing memory write references for efficient management of hybrid PCM and DRAM Memory[C]// 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems, July 25-27, 2011, Singapore. Piscataway: IEEE Press, 2011: 168-175.

[29] LEE M, DONG H K, KIM J, et al. M-CLOCK: migration-optimized page replacement algorithm for hybrid DRAM and PCM memory architecture[C]// ACM Symposium on Applied Computing, April 13-17, 2015, Salamanca, Spain, New York: ACM Press, 2015: 2001-2006.

[30] SHIN D J, PARK S K, KIM S M, et al. Adaptive page grouping for energy efficiency in hybrid PRAM-DRAM main memory[C]// ACM Research in Applied Computation Symposium, March 26-30, 2012, Riva del Garda (Trento), Italy. New York: ACM Press, 2012: 395-402.

[31] KHOUZANI H A, YANG C, HU J. Improving performance and lifetime of DRAM-PCM hybrid main memory through a proactive page allocation strategy[C]// Design Automation Conference, Jun 7-11, 2015, San Francisco, USA. Piscataway: IEEE Press, 2015: 508-513.

[32] WU Z L, JIN P Q, YANG C C, et al. APP-LRU: a new page replacement method for PCM/DRAM-based hybrid memory systems[M]. Springer: Network and Parallel Computing, 2014: 84-95.

[33] CHEN K, JIN P, YUE L. A novel page replacement algorithm for the hybrid memory architecture involving PCM and DRAM[M]. Springer: Network and Parallel Computing, 2014: 108-119.

[34] HU J, ZHUGE Q, XUE C J, et al. Software enabled wear-leveling for hybrid PCM main memory on embedded systems[J]. IEEE Transactions on Very Large Scale Integration Systems, 2013, 23(4): 599-602.

[35] PARK S K, MAENG M K, PARK K W, et al. Adaptive wear-leveling algorithm for PRAM main memory with a DRAM buffer[J]. ACM Transactions on Embedded Computing Systems, 2014, 13(4): 1-25.

[36] JIANG L, DU Y, ZHANG Y, et al. LLS: Cooperative integration of wear-leveling and salvaging for PCM main memory[C]// The 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks, Jun 27-30, 2011, Washington, D C, USA. Washington, DC: IEEE Computer Society, 2011: 221-232.

[37] CHEN C H, HSIU P C, KUO T W, et al. Age-based PCM wear leveling with nearly zero search cost[C]// Design Automation Conference, January 3-7, 2012, San Francisco, USA. New York: ACM Press, 2012: 453-458.

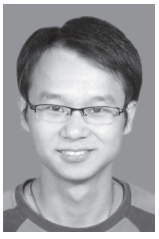
[38] SCHECHTER S, LOH G H, STRAUS K, et al. Use ECP, not ECC, for hard failures in resistive memories[J]. ACM Sigarch Computer Architecture News, 2010, 38(3): 141-152.

[39] 夏飞, 蒋德钧, 熊劲. 影响非易失性内存系统性能的因素分析[J]. 计算机研究与发展, 2014(S1): 25-31.
XIA F, JIANG D J, XIONG J. Evaluating and analyzing the performance of nonvolatile memory system[J]. Journal of Computer Research and Development, 2014(S1): 25-31.

[40] 何炎祥, 沈凡凡, 张军, 等. 新型非易失性存储架构的缓存优化方法综述[J]. 计算机研究与发展, 2015, 52(6): 1225-1241.
HE Y X, SHEN F F, ZHANG J, et al. Cache optimization approaches of emerging non-volatile memory architecture: a survey[J]. Journal of Computer Research and Development, 2015, 52(6): 1225-1241.

[41] 徐远超, 闫俊峰, 万虎, 等. 新型非易失存储的安全与隐私问题研究综述[J]. 计算机研究与发展, 2016, 53(9): 1930-1942.
XU Y C, YAN J F, WAN H, et al. A survey on security and privacy of emerging non-volatile memory[J]. Journal of Computer Research and Development, 2016, 53(9): 1930-1942.

作者简介



李鑫(1987-), 男, 博士, 南京航空航天大学计算机科学与技术学院讲师、硕士生导师, 中国计算机学会(CCF)会员, 主要研究方向为云计算、数据管理与分析、内存计算等。

陈璇(1996-), 女, 南京航空航天大学自动化学院本科生, 主要研究方向为云计算与大数据。

黄志球(1965-), 男, 博士, 南京航空航天大学计算机科学与技术学院教授、博士生导师, CCF杰出会员, 主要研究方向为嵌入式软件安全性、形式化验证技术、隐私保护等。

收稿日期: 2017-11-21
基金项目: 国家高技术研究发展计划("863"计划)基金资助项目(No. 2015AA015303); 江苏省自然科学基金资助项目(No. BK20160813)
Foundation Items: The National High Technology Research and Development Program of China(863 Program) (No. 2015AA015303), Jiangsu Natural Science Foundation(No. BK20160813)