

文章编号: 1000-5641(2014)05-0072-10

面向非易失内存的结构和系统级设计与优化综述

孙广宇¹, 舒继武², 王 鹏¹

(1. 北京大学 信息科学技术学院, 北京 100871; 2. 清华大学 计算机系, 北京 100084)

摘要: 当今各类计算机应用都进入一个飞速发展的阶段, 无论是“计算密集型”还是“存储密集型”应用都对存储系统的容量、性能以及功耗不断提出更高的要求. 然而, 由于传统内存工艺(DRAM)的发展落后于计算逻辑工艺(CMOS), 基于 DRAM 的内存设计逐渐无法满足这些设计需求. 同时, 基于 HDD 的外存性能与 DRAM 主存间的差距也逐渐增加. 而各种非易失存储工艺取得长足的进步, 为解决这一问题提供了新的机遇. 本文就近年来针对非易失内存的结构和系统级设计与优化的研究工作进行综述, 揭示非易失内存对存储系统的性能、功耗等都有明显的改善.

关键词: 非易失存储; 存储类内存; 存储结构; 文件系统; I/O 接口

中图分类号: TP303 **文献标识码:** A **DOI:** 10.3969/j.issn.1000-5641.2014.05.006

Survey of architecture and system level optimization for non-volatile main memory

SUN Guang-yu¹, SHU Ji-wu², WANG Peng

(1. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China;
2. Department of Computer Science and Technology, Tsinghua University, Beijing 100085, China)

Abstract: With the rapid improvement of modern computing applications, there is an increasing requirement of capacity, performance, and power consumption of memory system for both “computing-intensive” and “data-intensive” applications. However, main memory based on traditional DRAM technology cannot fully satisfy the requirement because the improvement of DRAM technology is slower than that of CMOS technology. Moreover, it becomes even worse since the performance gap between HDD based storage and DRAM based main memory keeps increasing at the same time. Recently, the substantial progress of various non-volatile memory technologies has provided an opportunity to mitigate this problem. This paper presents a survey of recent architecture and system level research work on non-volatile main memory. It shows that different

收稿日期: 2014-06

基金项目: 国家 863 课题(2013AA013201)

第一作者: 孙广宇, 男, 特聘研究员, 博士生导师, 研究方向为高性能存储系统与体系结构, 非易失存储系统. E-mail: gsun@pku.edu.cn.

通信作者: 舒继武, 男, 教授, 博士生导师, 研究方向为网络(云/大数据)存储系统, 存储可靠性, 非易失存储系统与技术. E-mail: shujw@tsinghua.edu.cn.

types of non-volatile main memory can help improve performance and reduce power consumption of memory system significantly.

Key words: non-volatile memory; SCM; memory architecture; file system, I/O interface

0 引 言

传统的存储系统由三个主要存储层次构成,按访存延时(latency)从低到高的顺序依次是缓存、主存和内存。然而,基于传统存储工艺的存储系统已经逐渐无法满足计算机系统对性能和功耗的需求。换言之,计算机系统的发展正面临“存储墙”带来的巨大挑战。该问题主要归结于两个方面的原因:一方面,随着多核/众核处理器的普及以及计算机系统规模的飞速增长,传统的“计算密集型”应用对存储系统的延时和容量的需求不断提升;另一方面,大数据时代的到来使得各种“数据密集型”应用不断涌现,进一步对存储系统的带宽和容量提出更高的要求。下面,本文将围绕新兴的“内存计算(in-memory computing)”应用,通过分析这类应用对内存的需求,强调引入“非易失内存(non-volatile memory)”的必要性。

随着移动互联网、物联网和云计算技术的飞速发展,大数据时代已经来临。每天,数以万计的智能设备和传感器每时每刻都在向后台系统传输着数据。如何将这些实时产生的数据加工成实时的信息,为企业提供实时决策的依据,创造出更大的价值?这已经成为 IT 业界面临的一个新的挑战 and 机遇。内存计算有望使数据计算的速度呈几何级的增长,更好地解决这个问题。

内存计算是指采用高性能的内存来对海量数据进行实时计算和处理,CPU 直接从内存而不是硬盘上读取数据,它比传统的基于磁盘的系统快几个数量级。内存计算非常适合处理需要实时获得分析结果的数据。随着内存硬件价格的下降、容量的提高,内存计算有望得到广泛应用。当前,学术界已经有不少文献在研究相关的内存存储系统和内存数据库技术,比如 PicoLo^[5],Apache Spark^[6]的前身 RDD^[7, 8, 9]、RAMCloud^[10, 11, 12]等。

SAP 公司指出,内存计算对内存容量的需求非常巨大,在多核场景下,需要达到 TB 级^[1]。而 IMEX research 公司的报告指出^[2],各种不同类型的应用负载有着不同的 I/O 特性(如图 1 所示^[2]),例如联机事务处理(OLTP)类应用具有很高的随机 I/O,对延迟的要求最高,而 Hadoop 类大数据应用属于批处理型 I/O,以顺序写为主,对吞吐的要求较高。

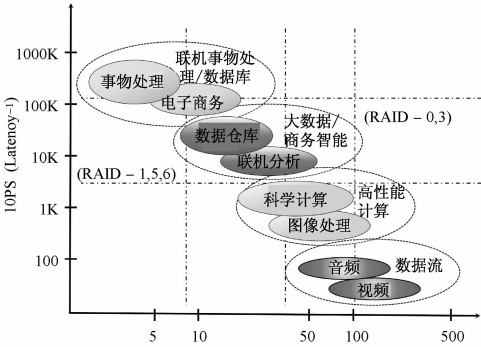


图 1 各种类型的应用负载的 I/O 特性

Fig. 1 I/O characteristics of various workloads

各种应用的 I/O 对于吞吐和延迟不同的需求决定了需要有不同的存储设备为之服务, 非易失存储的出现不仅能够显著降低引入大量内存产生的静态功耗, 同时丰富了存储层次结构, 有望填补 DRAM 内存和硬盘之间不断扩大的鸿沟, 如图 2^[2] 所示.

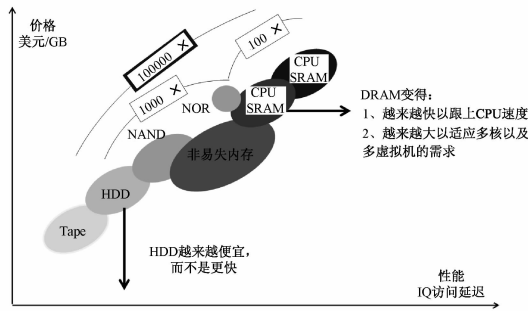


图 2 存储结构层次
Fig. 2 Memory hierarchy

在内存计算的场景下, 非易失内存有着广泛的应用场景. (1) NVM 有着比 NAND Flash SSD 更好的延迟和吞吐, 其 IO 性能比 SSD 好 100 倍以上; (2) NVM 的存储密度比 DRAM 更大, 因此在同等面积/内存插槽的情况下, 它能给多核环境下的 CPU 提供更多的数据; (3) 在前两点因素的共同作用下, 对于存储一定量的数据并满足一定性能需求的条件下, 使用 SCM 技术能降低所需要的机器数量. 例如: SAP 公司已经把非易失内存技术的应用研究作为它们内存计算技术研究的一个重点^[3], 此外, Virident 公司已经着手对 MySQL InnoDB 和 MemCached 用在 SCM 的情况进行了优化^[4], 使这些系统得到更好的利用.

1 基本知识

如前言所述, 维护已有 30 多年历史的基于 DRAM 和 HDD 的传统主存和外存结构, 已经成为面向新兴应用计算系统面临的主要挑战. 本章节首先介绍两类主要的非易失内存结构, 并通过对各种非易失存储器件的分析, 探讨适合这两类非易失内存结构的存储器件.

如图 3^[35] 所示, 本文所指的非易失内存主要分为两类: (1) 兼容传统主存控制器 (memory controller) 的非易失主存 (NVM main memory); (2) 连接在 I/O 总线上的存储级内存 (Storage-Class-Memory 或 SCM).

第一类非易失内存 (下文简称 NVM 主存) 的特点是可以直接替换传统的 DRAM 内存或者两者同时使用构成混合主存 (Hybrid Memory), 其优势是 NVM 主存由硬件进行控制, 只需针对 NVM 主存的特性对主存控制器进行适当的改进, 对上层应用透明, 无需引入系统级 (操作系统、文件系统等) 的改动. 这种非易失内存结构主要利用非易失存储器件低静态功耗的优势, 同时要满足能够匹配传统 DRAM 的性能需求, 并尽可能地提高存储密度.

第二类非易失内存 (下文简称 SCM), 是对介于传统 DRAM 主存和 HDD 外存之间的存储层次所使用的存储设备的统称. 相比于 HDD 外存, SCM 具备无运动部件、低延迟、高吞吐率的优势; 相比于 DRAM 主存, 具备非易失、低单位比特造价、低功耗的优势^[13].

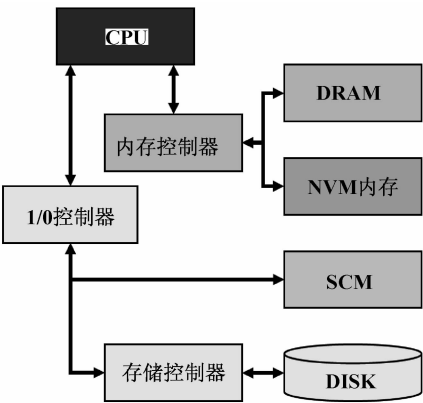


图 3 两类非易失内存的示意图

Fig. 3 Schematic of two nonvolatile memory types

通过两类非易失内存特性的对比,可以看出两类非易失内存对所采用的存储器件工艺提出不同的需求. 同时,针对网络搜索、内存计算、数据挖掘、移动计算等多种应用,即使同一类非易失内存也会针对实际应用进行不同侧重点的设计及优化. 因此,当今各种主流的闪存(FLASH)、铁电存储器(FeRAM)、磁存储器(MRAM)、相变存储器(PRAM)、阻变存储器(RRAM)等非易失存储器件都是制造非易失内存的潜在竞争者^[14]. 下文首先对集中非易失存储器件进行介绍,并分析其可能应用的场景.

闪存(FLASH)通过选择是否在浮栅中存放电荷进而存储数据. Flash 依据存储单元分为与非(NAND)和或非(NOR)两种结构. NOR 可以提供高达 100 MB/s 的读写速度,而 NAND 却可以达到数倍于 NOR 的存储密度,可以达到较低的生产成本. 同时,3D 堆叠技术将多层 Flash 存储单元堆叠在一起,为提高存储密度提供了新的途径^[15]. 目前,进入 22 nm 及以下的多比特与非闪存(MLC NAND Flash)已被业界用于生产固态硬盘(SSD),三星、闪迪等公司均发布 1T 的 MLC Flash. 相比于传统 HDD,NAND Flash 具备低延时、高吞吐、低功耗的优势,但是性能尚无法与 DRAM 相比,同时考虑到其擦写寿命问题,NAND Flash 通常适合于设计大容量,高吞吐的 SCM.

铁电存储器(FeRAM)是最早的非易失存储竞争者之一. 它由两端为金属电极中间为铁电材料构成的铁电容组成. 这种三明治结构功耗低、工作电压低,并可能较容易与 CMOS 工艺集成. 但是 FeRAM 的存储单元较大,制成温度高,刻蚀无斜坡的三明治结构非常困难,铁电也面临 3D 集成等诸多挑战. 考虑到 FeRAM 存储密度较低,通常适合在(超)低功耗嵌入式系统中用于非易失内存设计,并不适合面向新兴高性能应用的计算机系统.

磁存储器(MRAM)通过磁性材料的磁极方向保存数据. MRAM 提供较快的写性能、较容易的 CMOS 集成方式、高的耐用性. 目前 MRAM 的问题是写电流较大,电迁移现象随着尺寸缩小而严重,影响器件性能. 包括自旋矩传输磁存储器(STT-MRAM)、条带磁存储器(Racetrack)在内的多种 MRAM 变体试图降低写电流. 已经有 Everspin, Grandis 和日本电气(NEC)等公司成功展示了自旋矩传输磁存储器的商业化产品. STT-RAM 性能优于 DRAM,但是存储密度尚有一定差距,因此可以与 DRAM 进行混合主存设计. Racetrack 相对于 STT-RAM 存储密度有显著提高,是未来 NVM 主存设计的有力竞争者之一.

相变存储器(PRAM)利用硫化物晶态与非晶态间的巨大电阻差值来记忆数据. PRAM 可以提供高耐用性、快的读写速度、22nm 后良好的缩小潜力等. PRAM 的存储密度以及读操作性能都可以和 DRAM 相媲美,是制造 NVM 主存的另一个有力竞争者. 但是 PRAM 需要较大的写电流,在写操作性能方面低于 DRAM. 而且 PRAM 的擦写次数通常小于 108,因此如何提高写操作性能和寿命问题是 PCM 主存设计面临的主要挑战. 随着 PRAM 工艺的进展,多比特 PRAM 存储器件(MLC)也受到广泛关注. 基于 MLC 的 PRAM 设计可以提供和 NAND Flash 相媲美的存储密度,但是性能和寿命将进一步下降,因此,更加适用于 SCM 设计或者混合 NVM 主存设计.

阻变存储器(RRAM)利用多种材料在适当的电压下的不同电阻来存储数据. RRAM 按照材料可以分为绝缘体存储和固态电极存储. RRAM 通过横木交错(crossbar)结构可以提供更高的存储密度,并提供高于 Flash 数倍的访问速度. 实验表明 RRAM 只需要较小的复位电流(10 μ A),并可以在 85 摄氏度下工作 10 年以上^[16]. RRAM 当今面临的主要问题是工艺扰动较大,导致其存储芯片的良率(yield)较低. 近年来,RRAM 的制造工艺一直在不断进步,预期 RRAM 在 16 nm 及之后技术节点,可能成为 NVM 主存和 SCM 有力的竞争者.

综上所述,各种非易失存储的发展为 NVM 主存及 SCM 两类非易失存储的设计都提供了可行性. 下面两个章节中,我们将分别对近期 NVM 主存和 SCM 两个领域的部分代表性工作进行综述,在揭示非易失存储带来的性能及功耗优势的同时,说明在体系结构级和系统级两个层次需要进行的改进与优化.

2 NVM 主存体系结构级优化

传统主存通常采用 DRAM 工艺,如前文所述,DRAM 工艺同样存在静态功耗较高的问题. 因此研究人员提出使用 PCM 代替 DRAM 工艺. 表 1^[17]对基于 DRAM 和 PCM 的主存的部分参数进行了相应的对比.

表 1 DRAM 和 PCM 部分技术参数
Tab. 1 Technology parameters of DRAM and PCM

	DRAM	PCM
单元大小	6-8 F ²	4 F ²
访问延时	预充时间(tRCD, tRP)等较慢	预充时间(tRCD, tRP)等较快
读能耗	1. 17 pJ/位	2. 47 pJ/位
写能耗	0. 39 pJ/位	16. 82 pJ/位
写循环次数	10 ¹⁵	10 ⁸

可以看出,虽然 PCM 可以显著减少主存的静态功耗,其写操作同样存在高能耗和高延时的问题. 更重要的是,PCM 的擦写次数远低于 DRAM,因此需要考虑 PCM 主存寿命的问题. 本章我们主要介绍提高 PCM 能效的结构设计和优化技术,关于寿命的问题将在本文后续部分讨论.

2009 年, Ping Zhou, Benjamin Lee 和 Moinuddin Qureshi 分别同时提出采用 PCM 代替 DRAM 进行主存设计^[16-18]. 这些研究工作指出了基于 PCM 主存系统所存在的问题,提出了包括 bank buffer 在内的主存结构优化设计,并提出了两种简单有效的优化方法. 一种是

用 DRAM 作为 PCM 的缓冲(Buffer)减少对 PCM 的写操作,另一种是采用部分写(partial writes)策略,写入数据时和原始数据进行比较,只将差异部分写入。这两种方法对于改善基于 PCM 主存系统的性能有明显的效果。

随着 PCM 技术的发展,人们逐渐认识到多比特(Multi-level Cell)PCM 存储单元可以进一步提高主存的存储密度。但是,多比特 PCM 单元的“写操作”延时和能耗显著的增加极大阻碍了其在内存层次的应用。因此,针对这一情况,研究人员提出了大量的结构优化设计^[19-23]。例如,可以通过多比特与单比特 PCM 单元结合的方法在性能和容量之间寻求优化平衡;通过多级的写入及比较的方法,能够有效地提高“写操作”的能效;通过数据的选择性组合,可以提高 PCM 数据写操作的带宽;通过类似于 STT-RAM 缓存中使用的写暂停优化技术来提高 PRAM 写性能^[19-23]。

数据编码技术也是一种有效提高 PCM 写操作能效的方法。本文作者提出根据数据访问频率以及写入能量进行单比特和多比特 PCM 的数据编码优化技术^[24,25]。最近,其他研究人员提出采用 WoM-code 的数据编码技术来提高 PCM 写操作的性能和功耗^[26]。Wang-yuan Zhang 等人还针对工艺扰动对 PCM 主存的影响进行了结构级的优化设计^[27]。值得一提的是,除了 PCM 以外,Engin Ipek 等人提出 RRAM 也可以作为 DRAM 的替代品之一,除了相应的设计改动,在结构层次提出了数据复制等优化策略^[28]。

Jeremy Condit 等人则研究了软硬件的合理分工、协同优化问题。他们围绕字节可寻址的持久存储器(byte-addressable persistent memory, BPRAM)设计了 BPFS 相应的硬件架构和文件系统^[29]。其设计原则是:(1)BPRAM 可以直接被 CPU 访问,无需隐藏在 I/O 控制器后。他们将 BPRAM 直接放在内存总线上,使用 L1/L2 cache 代替了 DRAM buffer cache;(2)硬件架构上添加了 BPFS 所需的顺序性和原子性功能的支持,同时仍然保持 L1 和 L2 高速缓存的性能优势。在 BPFS 中,为了保证更新顺序提出了一种叫做 epoch barriers 的新机制,一个 cache line 被标识一个 epoch 号,并且修改了 cache 硬件来保证内存写回顺序总是按照 epoch 号顺序进行;(3)BPFS 利用 SCM 的字节可寻址和原地更新特性,使用了“短路影子分页”(short-circuit shadow paging)技术来支持对 SCM 原子化、细粒度、快速、一致的更新。作者评估了三种情形:运行在 DRAM 上的 BPFS,和运行在 RAM disk 上和传统硬盘上的 NTFS;然后作者使用微体系模拟器来评测 BPFS 在 PCM 上的性能。实验结果显示,在 DRAM 上的 BPFS 的性能是 RAM disk 上的 NTFS 的两倍。同时,BPFS 比传统的文件系统提供更强的可靠性保证。

3 SCM 接口及文件系统优化

本章节在系统接口和文件系统两个层次讨论如何针对 SCM 进行优化。首先,由于 SCM 无法使用传统 DRAM 或者 HDD 的控制器(即 memory controller 和 storage controller),与 SCM 相关的系统接口是急需解决的问题之一。其次,如何改进当前为 HDD 设计的文件系统从而更加高效地访问 SCM 是第二个关键问题。

3.1 SCM 接口设计与优化

Haris Volos 等人研究了针对 SCM 的文件系统接口优化。由于 SCM 使用了与当前的块设备完全不同的接口,并且 SCM 的访问延时远低于磁盘,因此,现有的根据磁盘设计的内核存储栈带来的高访问延迟开销限制了文件系统利用 SCM 技术的能力^[31]。他们认为需

要重新审视存储 I/O 栈。据此,他们提出了一个针对 SCM 的灵活的文件系统接口,称为 Aerie^[32]。如图 4^[32]所示,Aerie 具有分散的文件系统结构,它以运行时库的形式把 SCM 直接暴露给用户程序,使它们无需与内核交互就能够读写文件。Aerie 降低了操作系统内核的在文件访问中的参与度,内核仅仅是起到粗粒度的分配和保护作用。Aerie 的好处在于,可以使应用程序无需改动复杂的内核代码就可以通过修改文件系统接口做针对性的优化。Volos 等人用 Aerie 实现了一个通用的 POSIX 标准文件系统 PXFS,其性能与基于内核的实现相近(RamFS)或者更好(ext4)。他们还设计了一个专门的文件系统 FlatFS,通过把文件系统的抽象语义缩减成一个具有更低一致性保证的键/值存储(key/value store),FlatFS 的性能比一个内核文件系统高出 20% - 109%。因此,把文件系统功能分散到各个客户端程序能够带来更灵活的实现,同时极大地提高性能。

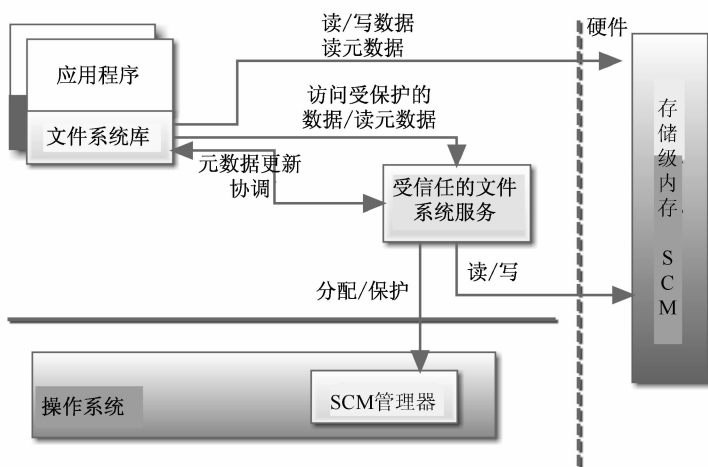


图 4 Aerie 的系统架构示意图

Fig. 4 System architecture of Aerie

尽管 IO 栈的调整与重构对存储系统性能的提升效果明显,但是仍然存在的文件系统语义对编程者而言依旧不够方便。因此,研究者希望利用 SCM 对所有需要持久化的数据都提供一个统一的编程接口,隔离其中的复杂实现,使 SCM 的使用更加高效。

面向磁盘开发的程序员都遵守着一条规则:不要频繁地手动更新持久化的数据,尽量使用数据库引擎来统筹处理。SCM 带来了实现快速的数据持久化的新机会,有望颠覆这条规则。除了文件系统接口优化之外,Haris Volos 等人还提出了针对 SCM 的编程接口,称为 Mnemosyne^[30]。Mnemosyne 提供轻量级的持久性存储器接口,使普通的内存数据结构可以容易地使用持久化事务来实现持久化。因此,程序员可以创建一个单一的、为内存设计的数据结构,而不用单独设计两种分别针对内存优化和针对持续性存储优化的数据结构。Mnemosyne 主要解决了两个问题:如何创建和管理这些内存、如何在故障出现时确保一致性。除此之外,Mnemosyne 提供直接修改持久型变量的编程原语,并通过一个轻量级的事务机制支持一致的更新。相比过去基于磁盘的持久性存储系统,Mnemosyne 通过细粒度的更新直接将数据写入到内存。与通过文件系统将内存页面写回磁盘的传统方式相比,Mnemosyne 减少了访问存储的延迟,提高了性能;相较于过去的持久对象系统,Mnemosyne 不

要求实现成C++ 对象,从而提供了更大的灵活性.

3.2 面向 SCM 的文件系统

最近几年,研究者提出了许多基于存储级内存(storage-class memory, SCM)而构建的文件系统.

Xiaojian Wu 等人提出了一个在虚拟地址空间中实现的简洁的文件系统 SCMFS^[33]. SCMFS 利用内存管理单元(MMU)将文件系统的地址映射到 SCM 的物理地址,形成了一个简单的地址空间布局. SCMFS 还利用了操作系统已有的内存管理模块来进行块管理,使得每个文件的虚拟地址空间总是连续的,简化了文件系统处理读写请求的过程. 同时,为了减少内存管理开销,SCMFS 采用了空间预分配机制,以及相应的垃圾回收机制. 作者在 Linux 下实现了一个原型,他们认为 SCMFS 的简洁设计不仅简化了其实现,而且通过实验发现这样的设计还获得了良好的性能(尤其是对于小的 I/O 请求).

Jaemin Jung 等人考虑到了 SCM 当前阶段存储容量不能满足需求的问题,提出了由 SCM 和 NAND 闪存一起构成的混合文件系统 FRASH^[34],即将大容量的 NAND 闪存和 SCM 一起构成一个文件系统是当前一个合理的选择. 在 FRASH 中,SCM 作为主存和存储的两方面特性同时得到了利用:一方面,SCM 可以被映射到主存储器的地址空间,此时它可以对落在相应的地址范围内的数据提供支持字节寻址(byte-addressability)的非易失性存储服务;另一方面,SCM 可以被当作块设备来使用,在这种情况下 I/O 速度会显著变快. 我们知道,日志式文件系统(log-structured file system)在 NAND 闪存上有着广泛的应用,它由内存中和磁盘上的两部分数据结构构成. 如图 5^[34]所示,FRASH 把日志式文件系统内存中的关键数据结构放在 SCM 里. 利用 SCM 的非易失性,FRASH 把挂载延迟降低了一个数量级,克服了日志式文件系统的一个主要缺点. 与此同时,通过在 SCM 中保存目录快照和文件树,FRASH 的可靠性得到提高,能够更好地应对突发故障. 此外,FRASH 还将原先位于磁盘的部分文件系统结构迁移至 SCM 中,从而支持对文件系统对象和元数据页面的字节寻址,显著地提高 I/O 性能. 值得一提的是,FRASH 还采用了“挂载时复制”技术来克服 DRAM 和 SCM 之间的访问延迟差距带来的不利影响.

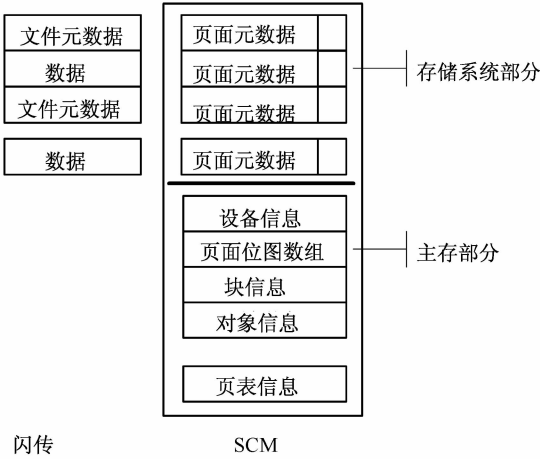


图 5 FRASH 文件系统结构示意图
Fig. 5 Illustration of FRASH file system

4 总 结

综上所述,针对内存计算等新兴应用,基于非易失存储器件设计的非易失内存可以提高存储系统的性能并降低功耗。根据非易失内存所处的存储层次以及应用的不同需求,可以将非易失内存分为 NVM 主存和 SCM 两类设计。对于 NVM 主存,由于其受硬件控制器管理,主要需要在体系结构层次上进行优化;而对于 SCM,需要同时在接口和文件系统两个层次方面进行设计和优化,从而有效地发挥非易失内存的优势。

[参 考 文 献]

- [1] SAP. In memory computing technology [R/OL]. 2013. http://fm.sap.com/data/UPLOAD/files/SAP_In-Memory_Computing_Technology_.pdf.
- [2] IMEX Research. System Architecture for In-Memory Database [R/OL]. 2013. <http://imexresearch.com/IMEX-Presentation/InMemoryComputing.pdf>.
- [3] SAP. Future storage memory technologies [R/OL]. 2013. <http://www.sap-innovationcenter.com/2013/11/21/future-storage-memory-technologies/>.
- [4] Virident Systems. Using Storage Class Memory for Extreme Performance In-Memory Computing [R/OL]. 2013. http://www.percona.com/files/presentations/ppc2009/PPC2009_using_storage_class_memory.pdf.
- [5] POWER R, LI J. Piccolo: Building Fast, Distributed Programs with Partitioned Tables[C]//OSDI, 2010, 10: 1-14.
- [6] APACHE SPARK. Spark [R/OL]. 2014. <http://spark.apache.org/>.
- [7] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing[C]//Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USENIX Association, 2012: 2-2.
- [8] ZAHARIA M, DAS T, LI H, et al. Discretized streams: Fault-tolerant streaming computation at scale[C]//Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles. ACM, 2013: 423-438.
- [9] XIN R S, ROSEN J, ZAHARIA M, et al. Shark: SQL and rich analytics at scale[C]//Proceedings of the 2013 international conference on management of data. ACM, 2013: 13-24.
- [10] OUSTERHOUT J, AGRAWAL P, ERICKSON D, et al. The case for RAMCloud[J]. Communications of the ACM, 2011, 54(7): 121-130.
- [11] ONGARO D, RUMBLE S M, STUTSMAN R, et al. Fast crash recovery in RAMCloud[C]//Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM, 2011: 29-41.
- [12] RUMBLE S M, KEJRIWAL A, OUSTERHOUT J K. Log-structured memory for DRAM-based storage[C]//FAST. 2014: 1-16.
- [13] BURR G W, KURDI B N, SCOTT J C, et al. Overview of candidate device technologies for storage-class memory [J]. IBM Journal of Research and Development, 2008, 52(4.5): 449-464.
- [14] FREITAS R F, WILCKE W W. Storage-class memory: The next storage system technology[J]. IBM Journal of Research and Development, 2008, 52(4.5): 439-447.
- [15] KIM K. From the future Si technology perspective: Challenges and opportunities[C]//Electron Devices Meeting (IEDM), 2010 IEEE International. IEEE, 2010: 1.1.1-1.1.9.
- [16] ZHOU P, ZHAO B, YANG J, et al. A durable and energy efficient main memory using phase change memory technology[C]//ACM SIGARCH Computer Architecture News. ACM, 2009, 37(3): 14-23.
- [17] LEE B C, IPEK E, MUTLU O, et al. Architecting phase change memory as a scalable dram alternative[J]. ACM SIGARCH Computer Architecture News, 2009, 37(3): 2-13.
- [18] 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.
- [19] JIANG L, ZHANG Y, CHILDERS B R, et al. FPB: fine-grained power budgeting to improve write throughput of multi-level cell phase change memory[C]//Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2012: 1-12.
- [20] JIANG L, ZHAO B, ZHANG Y, et al. Improving write operations in MLC phase change memory[C]//High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on. IEEE, 2012: 1-10.
- [21] JOSHI M, ZHANG W, LI T. Mercury: A fast and energy-efficient multi-level cell based phase change memory system[C]//High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on. IEEE, 2011: 345-356.
- [22] QURESHI M K, FRANCESCHINI M M, LASTRAS-MONTAÑO L A, et al. Morphable memory system: a robust architecture for exploiting multi-level phase change memories[C]//ACM SIGARCH Computer Architecture News. ACM, 2010, 38(3): 153-162.
- [23] QURESHI M K, FRANCESCHINI M M, LASTRAS-MONTAÑO L A. Improving read performance of phase change memories via write cancellation and write pausing[C]//High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on. IEEE, 2010: 1-11.
- [24] WANG J, DONG X, SUN G, et al. Energy-efficient multi-level cell phase-change memory system with data encoding[C]//Computer Design (ICCD), 2011 IEEE 29th International Conference on. IEEE, 2011: 175-182.
- [25] SUN G, NIU D, OUYANG J, et al. A frequent-value based PRAM memory architecture[C]//Proceedings of the 16th Asia and South Pacific Design Automation Conference. IEEE Press, 2011: 211-216.
- [26] ZHANG X W, JANG L, ZHANG Y, et al. WoM-SET: Low power proactive-SET-based PCM write using WoM code[C]//Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on. IEEE, 2013: 217-222.
- [27] ZHANG W, LI T. Characterizing and mitigating the impact of process variations on phase change based memory systems[C]//Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture. ACM, 2009: 2-13.
- [28] IPEK E, CONDIT J, NIGHTINGALE E B, et al. Dynamically replicated memory: building reliable systems from nanoscale resistive memories[C]//ACM SIGARCH Computer Architecture News. ACM, 2010, 38(1): 3-14.
- [29] CONDIT J, NIGHTINGALE E B, FROST C, et al. Better I/O through byte-addressable, persistent memory [C]//Proceedings of the ACM SIGOPS 22nd symposium on operating systems principles. ACM, 2009: 133-146.
- [30] VOLOS H, TACK A J, SWIFT M M. Mnemosyne: Lightweight persistent memory[C]//ACM SIGARCH Computer Architecture News. ACM, 2011, 39(1): 91-104.
- [31] VOLOS H, PANNEERSELVAM S, NALLI S, et al. Storage-class memory needs flexible interfaces[C]//Proceedings of the 4th Asia-Pacific Workshop on Systems. ACM, 2013: 11.
- [32] VOLOS H, NALLI S, PANNEERSELVAM S, et al. Aerie: flexible file-system interfaces to storage-class memory[C]//Proceedings of the Ninth European Conference on Computer Systems. ACM, 2014: 14.
- [33] WU X, REDDY A L. SCMFS: a file system for storage class memory[C]//Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. ACM, 2011: 39.
- [34] JUNG J, WON Y, KIM E, et al. FRASH: Exploiting storage class memory in hybrid file system for hierarchical storage[J]. ACM Transactions on Storage (TOS), 2010, 6(1): 3.
- [35] BURR G W. Storage Class Memory: Towards a disruptively low-cost solid-state non-volatile memory [R/OL]. IBM Almaden Research Center. [2013-01-12]. <http://researcher.ibm.com>.