持久性内存文件系统研究综述

徐远超　闫俊峰　杨璐
首都师范大学 信息工程学院

引言

近年来，随着半导体和新材料技术的发展，出现了很多新型的非易失存储介质（Non-Volatile Memory, NVM），比如PCM、STT-RAM、3D XPoint等，给现有的存储体系结构带来了新的契机。这些新型存储介质具有功耗低、访问速度快、存储密度大、字节可寻址以及非易失等优点，不仅有效缓解了内存计算（in-memory computing）中现有内存扩展能力不足的问题，而且还可用于持久性存储，能够提供比闪存更大的带宽和IOPS，大幅提高系统的性能。

然而，研究表明，即使所有的存储介质均使用NVM，二级存储结构的延迟和功耗也要高出单级存储结构4倍，原因在于单级存储结构消除了数据在两级存储介质之间的传输，有效降低了数据传输的开销和功耗。然而，要实现内存与外存的真正融合，却任重而道远，因为目前所有的系统软件（尤其是操作系统）都是针对二级存储模型设计的，即内外存分别通过虚拟内存模块和文件系统模块来管理。内存和外存的融合，需要包括编程模型、编译模型、内存管理、文件系统在内的整个软件生态系统的革新，工程浩大，短期来看，通过轻量级的文件系统来管理NVM仍然是较为现实的选择，这样可以兼容传统的文件系统编程接口。因此，面向新型非易失存储介质的文件系统得到了学术界和企业界的极大关注。

NVM带给OS的挑战

由于PCM等存在写入次数有限问题，因此，早期的研究聚焦在混合存储上，Mogul等人[2]讨论了在混合存储介质环境下，操作系统需要提供哪些支持。主要包括，由于读写延时的差异性，需要DRAM对数据进行缓冲以满足同步接口的时序约束，该缓冲区对操作系统是可见的，会对页迁移机制的实现产生影响；为了缓解耐用性问题，要求操作系统提供一些智能化的策略来决定什么页，以及何时应该迁移到NVM。Bailey等人[3]也讨论了NVM对操作系统的影响，包括，不需要页的交换，因为NVM本身就是低延迟、非易失的存储介质；页的粒度需要调整：还需要页来进行存储管理，比如分配，保护，存储空间的规划等，应尽量减小存储分片，减小页表结构的开销；主存和外存的保护机制，主存以页为单位进行保护；外存以文件为单位进行保护；主存和外存合并到一起，这两种保护机制势必需要统一；名字空间：每个进程都有自己的地址空间；文件系统也有自己的命名空间，这两个空间是否可以统一等。此外，NVM给操作系统带来的挑战还包括：1）需要重新优化I/O软件栈以降低开销；2）工作内存中非易失数据的安全和隐私问题；3）对于单级存储系统，是否延用面向虚拟内存和文件系统的分离的接口，仍然是一个开放的问题。短期来看，针对NVM进行文件系统优化，消除和减少不必要的工作，仍然是有现实意义的。

面向NVM的文件系统

2.1 降低一致性开销

Condit等人[6]设计了一个面向字节可寻址，持久性内存的文件系统BPFS。Ou等人[10]提出一种针对非易失内存系统进行优化的新型文件系统FCFS。

2.2 提供空间利用率

Nathan等人[13]提出一种面向NVM的压缩文件系统原型MRAMFS，来提高有限的NVM资源的利用率。

2.3 内存文件系统安全

Volos等人[15]提出Aerie，将文件系统的实现分为不可信的用户模式库（libFS）和可信的文件系统服务（TFS），用户库提供了访问文件的接口，包括命名和数据访问。系统服务通过确保元数据的完整性和同步提供了不可信程序之间的协作服务。

2.4 内存管理与文件系统融合

Wu等人[16]提出了一个在虚拟地址空间中实现的简洁的文件系统SCMFS。Sha等人[17]针对持久化内存系统提出SIMFS，该文件系统基于文件虚拟地址空间的框架。

2.5 提高文件系统并发性

Ou等人[20]提出了一种精细化I/O处理的持久性内存文件系统HiNFS。Xu提出了NOVA[23]N与LFS相比进行了很大改进。

研究展望

4.1 扩展性问题

Wang等人[30]评测了PMFS和PMBD以及使用movent指令的EXT4/RAMDISK文件系统，虽然对于小的测试程序单个文件的读写PMFS要好于ext4，但对于大的测试程序，PMFS要远差于ext4和ramdisk。说明PMFS的扩展性存在问题，初步分析是PMFS使用clflush和sfence来维护一致性的开销非常大，可考虑使用新型指令，如clwb等。

4.2 VM与FS融合

NVM同时具有字节寻址特性和非易失特性，既可以当成传统的内存管理，也可以当成传统的外存管理。然而，如果当成传统的外存使用文件系统管理，无论是命名、索引还是访问，都很难发挥NVM的低延迟优势。如果当成内存使用虚拟内存来管理，又无法对文件进行管理。因此，研究方向之一是基于NVM的对象存储，使用扁平的命名空间，使用MMU等内存管理机制对文件对象进行管理。

4.3 分布式系统

在分布式环境下，NVM的部署也带来一些挑战，因为网络的传输延迟超过了NVM的写延迟[31]，分布式环境下的副本容错机制以及事务机制都需要进行改进，尽量减小传输延迟带来的影响。比如，基于RDMA协议分布式事务。

参考文献

[1] Mogul, Jeffrey C., Eduardo Argollo, Mehul A. Shah, and Paolo Faraboschi. "Operating System Support for NVM+ DRAM Hybrid Main Memory." In HotOS. 2009.
[2] 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.
[3] Lee E, Yoo S, Jang J E, et al. Shortcut-JFS: A write efficient journaling file system for phase change memory[C]//012 IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST). IEEE, 2012: 1-6.
[4] Dulloor S R, Kumar S, Keshavamurthy A, et al. System software for persistent memory[C]//Proceedings of the Ninth European Conference on Computer Systems. ACM, 2014: 15.
[5] Lee E, Jang J, Bahn H. DTFS: exploiting the similarity of data versions to design a write-efficient file system in phase-change memory[C]//Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM, 2014: 1535-1540.
[6] Ou J, Shu J. Fast and Failure-Consistent Updates of Application Data in Non-Volatile Main Memory File System[C]//Proceedings of the 32st Symposium on Mass Storage Systems and Technologies (MSST), IEEE , 2016.
[7] Zheng S, Huang L, Liu H, et al. HMVFS: A Hybrid Memory Versioning File System[C]//Proceedings of the 32st Symposium on Mass Storage Systems and Technologies (MSST), IEEE , 2016.
[8] Edel N. MRAMFS: A Compressing File System for Byte-Addressable Non-Volatile RAM[R]. Technical Report UCSC-SSRC-11-02, University of California, Santa Cruz, 2011.
[9]Protected and Persistent RAM Filesystem. PRAMFS [EB/OL]. [2016-07-08] http://pramfs.sourceforge.net/.
[10]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.
[11] Wu X, Qiu S, Narasimha Reddy A L. SCMFS: A file system for storage class memory and its extensions[J]. ACM Transactions on Storage (TOS), 2013, 9(3): 7.
[12] Sha E H M, Chen X, Zhuge Q, et al. Designing an efficient persistent in-memory file system[C]//Non-Volatile Memory System and Applications Symposium (NVMSA), 2015 IEEE. IEEE, 2015: 1-6.
[13] Kannan S, Gavrilovska A, Schwan K. pVM: persistent virtual memory for efficient capacity scaling and object storage[C]//Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016: 13.
[14]Ou J, Shu J, Lu Y. A high performance file system for non-volatile main memory[C]//Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016: 12.
[15] Sehgal P, Basu S, Srinivasan K, et al. An empirical study of file systems on nvm[C]//2015 31st Symposium on Mass Storage Systems and Technologies (MSST). IEEE, 2015: 1-14.
[16] Xu J, Swanson S. NOVA: a log-structured file system for hybrid volatile/non-volatile main memories[C]//14th USENIX Conference on File and Storage Technologies (FAST 16). 2016: 323-338.
[17] Josephson W K, Bongo L A, Flynn D, et al. DFS: a file system for virtualized flash storage[C]//Proceedings of the 8th USENIX conference on File and storage technologies. USENIX Association, 2010: 7-7.
[18] Min C, Kim K, Cho H, et al. SFS: random write considered harmful in solid state drives[C]//Proceedings of the 10th USENIX conference on File and Storage Technologies. USENIX Association, 2012: 12-12.
[19] Chen J, Wei Q, Chen C, et al. FSMAC: A file system metadata accelerator with non-volatile memory[C]//2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST). IEEE, 2013: 1-11.
[20] Qiu S, Reddy A L N. NVMFS: A hybrid file system for improving random write in nand-flash ssd[C]//2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST). IEEE, 2013: 1-5.

XUYUANCHAO@CNU.EDU.CN