大数据环境下的存储系统构建：挑战、方法和趋势

陈游旻，李飞，舒继武
清华大学计算机科学与技术系，北京 100084

摘要
互联网规模的迅速扩展促使全球数据总量呈现爆炸式的增长。物联网、电子商务等新的应用对数据存储及处理的实时性提出了更高的要求，迫切需要结合新型存储介质，以构建大规模、高性能存储系统。分别从闪存存储、持久性内存存储两种存储系统构建方案出发，详细阐述了其各自面临的挑战，并总结了现有的解决方案。最后，展望了未来数据中心及存储系统构建的若干发展趋势。

关键词
存储系统；闪存；非易失内存

中图分类号：TP316.4 文献标识码：A doi: 10.11959/j.issn.2096-0271.2019030

Building storage systems in big data era: challenges, methods and trends

CHEN Youmin, LI Fei, SHU Jiwu
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract
The rapid expansion of the Internet has led to an explosive growth in global data. New applications, such as the Internet of things and e-commerce, demand higher requirements for the latency and performance of data storage and processing. Thus, big data is not only more “big” but also more “fast”, and there is an urgent need to combine new storage media to build large-scale, high-performance storage systems. Two storage system design schemes were started, namely, flash storage and persistent memory storage. Their respective challenges were illustrated in detail, and the existing solutions were summarized. Finally, the future trends in the construction of data centers and storage systems were looked into.

Key words
storage system, flash memory, non-volatile memory

1 引言

互联网规模的迅速扩张促使全球数据呈现爆炸式增长、海量聚集的特点，大数据逐步走向信息化发展的新阶段。近年来，社交媒体、物联网等技术的迅速发展导致了大量非结构化数据的出现，从海量数据中提取有价值信息的难度越来越大。因此，大数据不止更“大”，还要更“快”。基于传统磁盘的大数据平台已经难以应对新应用日益增长的数据存储与处理需求，大数据技术正在发生着以下变化。

闪存（flash memory）逐渐替代磁盘，用于构建大规模存储系统。在存储系统中，磁盘自1956年被发明以来，长期居于外存的主导地位。近年来，随着闪存制造与加工工艺的逐渐成熟，闪存设备已经开始在个人电脑与移动设备上得到普及，并将在数据中心得到大规模的应用。据标准性能评估组织（Standard Performance Evaluation Corporation，SPEC）调查[1]，闪存设备在数据中心的使用比例从2012年的8%增长到了2017年的27%，预计到2020年，闪存设备的使用比例将达到47%。2018年，英特尔公司推出了基于4层单元（quad-level cell，QLC）和三维堆叠技术的消费级固态盘（solid state drive，SSD），在实现高性能、高可靠性的同时，进一步降低了固态盘的价格，提供了更大的存储容量。与磁盘相比，闪存具有体积小、能耗低、带宽高、时延低、抗震性强、可靠性高等特点。正因为如此，研究人员着力于构建大规模闪存存储系统，以充分发挥闪存优势，适应大数据环境的发展，如清华大学提出构建开放通道闪存系统等。闪存存储正发生着巨大变革。

随着内存价格的日益低廉，内存计算逐步成为人们关注的热点。电子商务、物联网、自动驾驶等应用对数据管理的时效性提出了新的需求。例如，一个典型的网页服务需要在数毫秒之内访问数据仓库数千次，用于响应单个用户请求[2]。内存计算依靠大容量内存，将待处理数据尽可能地全部放入内存中，从而实现高吞吐、高时效的数据存储与处理。然而，动态随机存取存储器（dynamic random access memory，DRAM）价格昂贵、能耗高、性能不稳定等缺陷，限制了内存计算被更广泛地应用。近年来出现了大量的新型非易失性随机存储介质（non-volatile memory，NVM），例如相变存储器（phase change memory，PCM）、阻变存储器（resistive ramdom-access memory，ReRAM）等，它们具有价格低廉、容量大、能耗低、性能与DRAM相当等特点，更重要的是，在系统发生断电故障时，其存储的数据不会丢失。这些优良的特性正推动着研究人员构建基于持久性内存的内存计算平台，如惠普实验室推出的The Machine、加州大学伯克利分校的FireBox等，内存存储与计算正面临着巨大变革。

针对大数据存储技术面临的巨大变革，本文将从闪存存储、持久性内存存储两种存储系统构建方案出发，详细阐述其各自面临的挑战以及解决方案，最后，展望未来数据中心及存储系统构建的若干发展趋势。

2 告别硬盘：闪存存储系统的构建

存储设备从机械式部件发展至电子式部件是计算机发展中的重大变革，也是计算机发展的趋势。**表1**比较了当前磁盘设备与闪存设备的存储性能。与磁盘设备相比，闪存设备带宽提高了1个数量级，时延

表1 闪存设备与磁盘设备的性能对比

设备型号	接口类型	带宽/(MB/s)		时延/ms		IOPS	
		读	写	读	写	读	写
希捷Savvio 15K.3	SAS 6 Gbit/s	202	202	2	2	543	428
Intel S3710	SATA 3 Gbit/s	550	520	0.055	0.066	85×10^3	45×10^3
Intel P3700	PCIe 3.0 x4	2 800	2 000	0.02	0.02	460×10^3	175×10^3

降低了2个数量级，每秒的输入输出次数（input/output operations per second，IOPS）提高了近3个数量级。当前存储系统多基于磁盘特性进行设计，极少考虑其他存储介质的特性。随着闪存技术的广泛应用，如何在大数据环境下高效利用闪存并构建适合于闪存的存储系统，值得关注与深入思考。

2.1 闪存与固态盘

闪存是一种电子式、可擦除、可编程、非易失的存储器件。与机械式的磁盘相比，闪存具有体积小、能耗低、带宽高、时延低、抗震性强、可靠性高等特点。在大容量固态盘中，NAND闪存是主要的存储介质，分为单层单元（single-level cell，SLC）、多层单元（multi-level cell，MLC）、3层单元（triple-level cell，TLC）与QLC，分别表示每个闪存单元记录1个、2个、3个和4个比特数。相比于传统磁盘介质，闪存主要具有以下几个独特性质。

● 写前擦除。闪存单元具有单向可编程的特性，对于闪存页的写入操作，闪存需要将数据写入已擦除的页面中，这被称为写前擦除，也被称为不可覆盖写。

● 读写擦粒度不同。在闪存设备中，闪存页是执行读写操作的基本单位，闪存块是执行擦除操作的基本单位。一个闪存页的容量一般为512 B~16 KB，一个闪存块中一般包含64~512个闪存页。此外，闪存设备的读、写、擦除操作的性能不同，闪存页的读操作平均时延为25 ms、写操作平均时延为200 ms，闪存块的擦除操作平均时延为1.5 ms[3]。

● 磨损寿命有限（耐久性）。闪存单元能承受的擦写操作次数有限。经过一定数量的擦写操作后，闪存单元不能可靠地存储数据状态，这一过程被称为单元磨损。闪存单元经历的擦写次数用于衡量闪存的擦写寿命，也被称为耐久性（endurance）。随着闪存单元中比特位的增加（从SLC到QLC），闪存设备的每比特价格在降低，耐久性问题更加严峻。

固态盘是由闪存存储单元组成的闪存存储设备，其内部存在不同级别的I/O并发访问能力，被称为内部并发特性。SSD采用闪存转换层（flash translation layer，FTL）对闪存的读写擦操作进行管理，并向软件系统提供与传统磁盘相同的读写接口，其主要功能包括地址映射、垃圾回收、磨损均衡、ECC校验、坏块管理等。地址映射机制维护了一个地址映射表，用于将主机端I/O请求的逻辑地址映射到闪存设备中的物理地址。由于闪存具有“写前擦除”的特性，FTL采用“异地更新”的方式对数据进行更新。FTL将新版本的数据通过地址映射写入空闲的闪存页中，并将存有旧版本数据的闪存页标记为无效。当SSD中的空闲闪存块数量低于预定义的阈值时，FTL会对SSD进行垃圾回收操作。在FTL的垃圾回收过程中，有效数据的移

动引入了额外的写入量，占用了闪存设备的有效带宽，加快了设备的磨损，这一问题被称为闪存设备的写放大问题。为了延长闪存设备的使用寿命，FTL采用动态或静态的磨损均衡策略，尽量使擦写操作均匀地分布在所有的闪存块上。在SSD中，每个闪存页除了数据区外，还保留一块带外空间（out of band，OOB），利用OOB对闪存页中存储的数据进行错误检查与纠正（error correcting code，ECC）。当某个闪存块无法可靠地保存数据时，FTL会将该块标记为坏块，不再使用。

2.2 闪存固态盘存储系统的问题

在FTL的帮助下，现有的存储系统可以无缝地运行在具有FTL的SSD之上，无须进行软件的修改。基于闪存固态盘的存储系统架构如**图1**所示。

SSD通过串行高级技术附件（serial advanced technology attachment，SATA）、高速串行计算机扩展总线标准（peripheral component interconnect express，PCIe）或非易失性内存主机控制器接口规范（non-volatile memory express，NVMe）硬件接口与主机端连接，将自身抽象成通用块设备供上层存储软件使用；内核中的文件系统运行在抽象出的块设备之上，并向用户态的应用程序（例如数据库）提供文件访问接口；FTL负责对闪存设备的特性进行管理，并将上层存储软件的I/O请求转换成对闪存页的读、写操作。FTL的应用加速了闪存设备的普及与推广，但也阻碍了现有存储系统发挥闪存设备在性能、寿命上的潜力，主要体现在以下几个方面。

- 层次间功能存在重叠与干扰。如**图1**所示，闪存固态盘存储系统中不同层次间存在功能冗余的问题，例如，在FTL、文件系统与上层数据库中都存在存储空间管理、数据地址映射或索引、垃圾回收等类似功能。这些冗余功能不仅会导致I/O处理上的低效，还会互相干扰，影响系统的性能和闪存设备的使用寿命。
- 系统软件无法感知闪存特性。FTL将闪存设备抽象成通用块设备，导出到主机端，屏蔽了上层存储软件对闪存设备特性的感知，阻碍了存储软件的定向优化。现

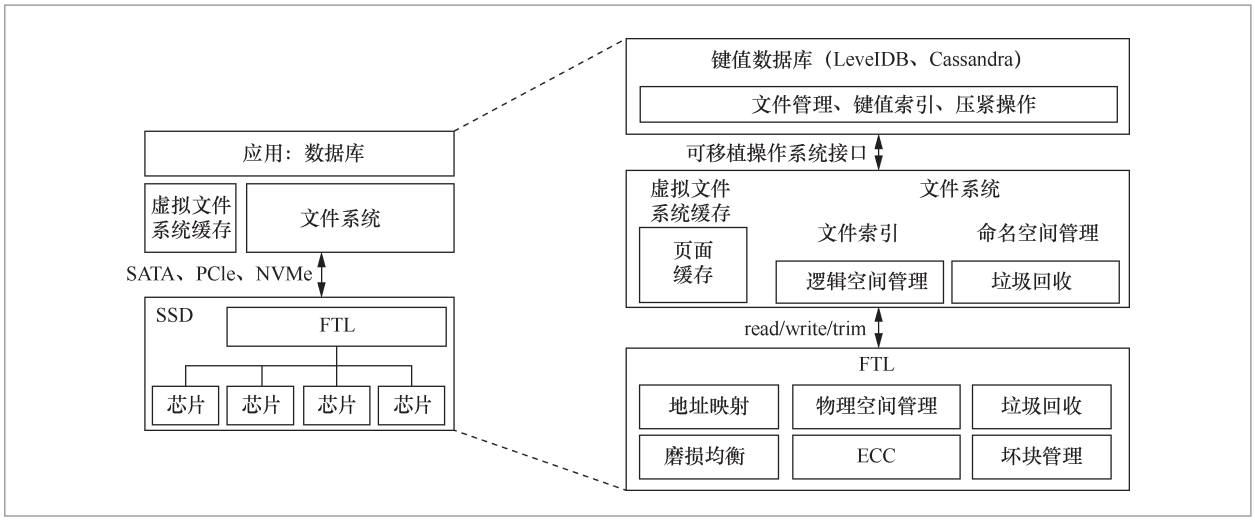


图1 基于闪存固态盘的存储系统架构

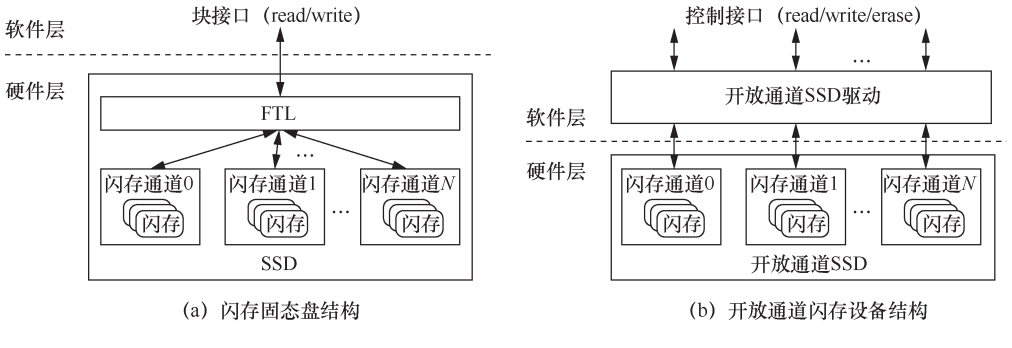
有的存储软件是基于磁盘设备进行设计与优化的，不能充分发挥闪存设备的优势，也无法弥补闪存设备的劣势。

● 硬件管理缺乏软件语义。上层存储软件通过FTL导出到块设备接口，将I/O请求发送到SSD上。块设备接口屏蔽了上层存储软件的语义信息，导致FTL在请求处理上效率低、产生额外的垃圾回收开销与写放大。例如，FTL在为写请求分配闪存物理空间时，由于缺乏写入数据的属性（如元数据或文件数据）与热度信息，无法有效地对写入数据进行布局优化。冷热数据可能会被分配到同一个物理闪存块中，导致垃圾回收的开销增大。

2.3 基于开放通道闪存设备的存储系统构建方法

近年来，一种新型闪存架构——开放通道（open channel）闪存架构得到了工业界与学术界的广泛关注[4-8]，为解决闪存固态盘存储系统面临的问题提供了思路。如**图2**所示，开放通道闪存架构在SSD的基础上，移除了设备端FTL，消除了闪存固态盘存储系统中的功能冗余。开放通道闪存架构将闪存设备的内部信息（如设备的硬件拓扑结构、闪存通道数量、闪存块大小、闪存页长度等）与控制接口（如读操作、写操作、擦除操作等）全部导出到主机端，由存储软件直接对闪存设备进行管理，打破了原有的感知屏蔽与语义隔离。存储软件能够根据自身的I/O特征与闪存特性进行软硬件协同设计与优化，充分发挥闪存设备的性能潜力，降低了设备的磨损。

开放通道闪存架构对云计算与数据中心具有重要意义。通过使用开放通道闪存设备，上层软件能够实现闪存通道级别的I/O隔离与并发控制[9]，能够对设备内的垃圾回收时机进行控制[10]，能够对闪存通道中的I/O请求按照语义优先级进行调度[7]。开放通道闪存架构的这些优势能够帮助存储系统实现可预测的I/O时延，降低分布式系统中的尾延迟（tail latency），优化云计算环境中的服务质量（quality of service，QoS）。目前，百度公司已经在其存储系统中部署了超过3 000块的开放通道SSD，用于网页和图像的存储服务[6]；阿里巴巴公司发布了自研的开放通道闪存设备AliFlash V3，并已经上线运行；谷歌、微软、脸书以及亚马逊都开始在数据中心应用开放通道闪存设备[11]，以降



（a）闪存固态盘结构　　（b）开放通道闪存设备结构

图 2 闪存设备架构比较

低I/O时延，提高服务质量。

虽然开放通道闪存架构具有上述优点，但是它也给存储系统的设计带来了挑战。首先，开放通道闪存设备的接口与块设备不同，它不兼容现有的基于块设备设计的存储软件与内核I/O栈（如缓存机制、I/O调度机制等）；其次，因为开放通道闪存设备移除了设备端的FTL，所以上层的存储软件需要对闪存设备的底层功能进行管理，例如坏块管理、磨损均衡、ECC校验等，这在一定程度上增加了存储软件的设计复杂度；最后，开放通道闪存设备导出的硬件信息与控制接口为存储系统的设计与优化带来了新的思考维度，例如，如何降低闪存设备的软件管理开销、如何充分发挥闪存设备的内部并发性能、如何将软件的I/O特征与闪存的特性结合、如何利用闪存的特性对现有的存储机制进行优化等。下面分别从闪存管理架构、文件系统、键值存储系统、分布式对象存储系统等方面介绍开放通道闪存存储系统的构建方法。

（1）闪存管理架构

由于开放通道闪存设备移除了设备端的FTL，主机端软件需要对闪存设备的内部特性进行管理，例如磨损均衡、坏块管理等，这增加了存储软件的设计复杂度。现有的主机端闪存管理架构存在着接口功能单一、适用场景受限、不兼容现有的存储软件栈等问题，限制了软件的优化空间与开放通道闪存设备的应用范围。针对上述问题，可以在主机端对闪存设备的管理功能进行分解，提供对开放通道闪存设备的底层管理与设备抽象，以降低上层存储系统的设计复杂度；设计细粒度（如页粒度）的闪存控制接口，以扩大存储软件的优化空间和设备的适用场景。除此之外，还可以在该架构的基础上设计实现主机端闪存转换层，为开放通道闪存设备提供块存储的支持。

（2）文件系统

由于FTL的屏蔽，基于固态盘的文件系统无法感知底层闪存设备的耐久性、内部并发特性等，不仅不能针对闪存特性进行定向优化，功能上的冗余与冲突甚至会导致额外的垃圾回收开销和写放大问题。针对上述问题，基于开放通道架构，移除设备内FTL，将原有FTL功能集成到文件系统的存储管理部分，由文件系统直接管理闪存介质，消除功能上的冗余和冲突。考虑到闪存耐久性，根据对象语义与闪存特性重新设计存储机制，包括：利用闪存页的OOB记录反向索引等额外信息，以延缓索引与日志的刷写；利用闪存块/页状态转换的特性设计空闲空间管理机制；对不对齐的写操作采用拼接紧凑写的机制等。这样的设计能大幅降低由文件系统自身机制引入的额外元数据写入，相比于传统文件系统，能显著降低文件系统写放大系数，提升系统性能，延长闪存寿命。针对发挥闪存内部并发特性与文件系统管理机制存在冲突的问题，采用日志式结构将文件系统中的数据分段与闪存物理块对应。在空间分配上采用二维分配机制，综合考虑设备的并发特性和数据冷热程度，在发挥闪存设备并发特性的同时保证冷热程度不同的数据相互隔离。另外，利用文件系统的语义信息直接对闪存块进行垃圾回收，在文件系统层为每个闪存通道的I/O请求进行优化调度。这样的设计能充分发挥闪存的内部并发特性，显著提升系统的整体性能，并且能较好地控制系统的性能抖动问题。

（3）键值存储系统

在闪存固态盘上，采用日志合并树（log-structured merge tree，LSM-tree）的键值存储系统存在三重的功能冗余，这不仅降低了系统的I/O处理效率，影响系统性能，冗余功能间的相互干扰还会导致严重的写放大[12]问题。同时，闪存设

备与LSM-tree的特性没有得到充分的利用与针对性的优化，阻碍了闪存与LSM-tree性能的发挥。针对上述问题，基于开放通道闪存架构，利用LSM-tree日志式更新的特征，在用户态直接对开放通道闪存设备进行管理，绕过文件系统与FTL，消除原有架构中的多重功能冗余与语义隔离。在此基础上，根据闪存设备的硬件特性与LSM-tree的读写特征，对键值存储系统的存储机制进行软硬件的协同设计与定向优化。采用基于“超级块”的空间管理机制，降低文件索引的开销；采用可重建的静态数据布局机制，在发挥闪存内部并发性能的同时，保证了系统故障后的一致性；采用动态并发的压缩机制，通过限制后台写请求的并发度以降低对前台读请求的干扰。进一步地，根据软件的语义信息对用户态I/O栈机制（如缓存机制、I/O调度机制等）进行定向优化。

（4）分布式对象存储系统

在基于闪存固态盘的分布式对象存储系统中，对象存储需要使用日志机制保证数据更新的一致性，这种“两遍写”的一致性保障机制不仅影响了系统的性能，还增加了系统的写放大系数。闪存设备的异地更新特性天然地保存了数据的多副本，但是现有的闪存事务机制在对象事务的一致性更新时会产生很大的开销，不适用于分布式对象存储的场景。针对上述问题，基于开放通道闪存架构，根据对象和事务的语义与闪存异地更新的特性，设计适用于分布式对象存储特性的高效闪存事务机制，利用闪存设备的异地更新特性与带外存储空间，可为对象数据及其相关元数据提供低开销的一致性更新保障。在此基础上，使用多线程将没有依赖关系的事务并行提交到物理隔离的闪存块中，在发挥闪存设备内部并发性能的同时，降低事务间的干扰。通过感知事务的语义，对各个闪存通道上的I/O请求执行顺序进行协调，将属于同一个事务的I/O请求在同一时段进行处理，降低系统的平均响应时延。

2.4 小结

随着大数据环境下海量数据存取对容量与实时性要求的不断提高，研究者也不断地寻求机会突破外存带来的性能瓶颈。从磁盘到闪存固态盘，实现了从机械式到电子式的跨越；从闪存固态盘设备到开放通道闪存设备，则从软硬件协同设计的存储架构的角度，创造性地提出了闪存存储系统构建的新思路。上述系统构建的经验表明，基于开放通道闪存设备的存储系统能够结合闪存设备的物理特性消除传统固态盘层次间的功能冗余，打破了语义隔离，在提高系统性能的同时增加闪存使用寿命。

3 不止更快：持久性内存存储系统构建

本节首先以英特尔公司推出的Optane持久性内存（Optane DC persistent memory）为例，介绍NVM的相关特性，然后阐述基于非易失内存构建存储系统时面临的问题，最后介绍基于非易失内存构建本地和分布式存储系统的设计方法。

3.1 非易失内存

英特尔公司于2019年4月正式发布Optane持久性内存，这是一款大规模量产的非易失内存设备。Optane持久性内存可以像DRAM一样，直接通过内存接口与CPU互连，并被CPU以字节粒度访问。目前，英特尔公司已经推出3款相关产品，其单条

容量分别为128 GB、256 GB和512 GB。Optane持久性内存有两种操作模式，分别为内存模式（memory mode）和应用直访模式（application direct mode）。用户可以灵活地将Optane持久性内存设置为不同的操作模式，以满足不同应用程序的性能需求。在内存模式下，将DRAM用作Optane持久性内存的缓存，从而大幅扩展了内存容量。上述缓存管理模式完全由内存控制器接管，因此，内存模式对操作系统完全透明，上层应用程序在不做出任何修改的情况下可以享受大容量内存带来的性能优势。在应用直访模式下，操作系统将DRAM和Optane持久性内存看作两个彼此独立的内存池，应用程序具有直接管理Optane的能力。该模式精简了软硬件栈的复杂度，应用程序可以按照各自的需求优化持久性内存的使用方法，以获取更优的性能，与此同时，也为相应的系统软件开发带来了更大的难度。据英特尔公司称，SAP HANA引入Optane持久性内存后，能够将系统重启速度提升13倍，并节省39%的成本。

加利福尼亚大学圣地亚哥分校也在Optane持久性内存发布的第一时间公布了其详细的测试报告[13]。该测试报告列举了Optane持久性内存的基础性能参数以及在内存模式和应用直访模式下的性能测试结果。该报告显示，Optane持久性内存的随机读时延为305 ns，这相比于传统的SSD具有两个数量级的改观，但时延仍比DRAM长3倍。同时，Optane对访问模式较敏感，在顺序访问时，读时延仅比DRAM长两倍。另外，Optane具有不对称的读写带宽：其最大读带宽可以达到39.4 GB/s，并能随着线程数量的增加而扩展，然而，其最大写带宽仅为13.9 GB/s，4个线程就能占满Optane持久性内存的写带宽。

3.2 非易失内存在实际应用中面临的挑战

现有的计算机体系结构均包含了多种存储介质，例如CPU中的寄存器和多级缓存、DRAM主存、固态硬盘、磁盘等，这些存储介质的特点是容量越大，速度越慢，距离CPU越远，这种存储结构被称为“金字塔”存储。Optane持久性内存作为一种全新的存储介质，其性能接近于DRAM，且提供了持久性数据存储，因此，Optane不属于现有的金字塔存储的任何一个层级。Optane持久性内存硬件上的变化为存储系统软件的设计带来了一系列新的问题。

（1）一致性管理开销高

非易失内存提供了主存层次的数据持久性，而处理器的片上缓存系统依旧是易失性的，系统故障可能导致非易失主存上的数据处于不一致的中间状态。目前的64位机器仅支持8 byte的数据原子写入操作，系统设计者需要额外的日志机制保证数据的一致性，即在修改某数据之前，先将新版本或旧版本的数据写到日志区，作为备份用于故障后的数据恢复。然而，非易失内存具有读写不对称的特性，写操作带宽严重受限，因此日志机制会引入极高的持久化开销。此外，处理器缓存由硬件管理控制，大多数现代处理器会对主存写操作进行重排序以提升性能，这些优化手段会打乱数据持久化到非易失内存的顺序，在系统故障时可能导致数据不一致的问题。因此，系统设计者需要通过额外的硬件刷写指令（如clflush、clflushopt等）按顺序强制实现数据的持久化。然而，这些硬件刷写指令开销极高。参考文献[14]指出，随着NVM的发展，预计存储系统的软件开销占比将高达94.09%。

(2)低效的操作系统抽象

操作系统将应用程序进行隔离，运行在用户态，而让内核服务程序运行在具有更高权限的内核态，用于硬件管理和抽象。应用程序通过系统调用访问内核服务程序（如文件系统等），进而与硬件设备进行交互，通过这种抽象机制，不同的应用程序彼此隔离，从而提供了更高的安全性。然而，系统调用过程将会引发一系列的现场保存与恢复、缓存逐出等额外开销，这使得在内核态管理持久性内存的开销变得更多。另外，通过内核态文件系统管理持久性内存空间时，Linux内核还在文件系统之上统一抽象了一层虚拟文件系统（virtual file system，VFS），在该层次，操作系统增加了粗粒度的锁管理机制和DRAM缓存系统。由于持久性内存和DRAM具有非常接近的性能，因此DRAM缓存不能再像在传统外存中一样发挥作用，进而在性能和扩展性上对文件系统的高效性产生极大的制约。

(3)分布式软件栈臃肿

为兼顾兼容性，现有的大多数分布式系统软件采用了模块化的设计，将分布式软件部署在本地文件系统之上。这种架构方式会引入一系列的冗余复制操作。例如，在应用程序读取数据时，数据需要从本地文件系统镜像分别复制到内核页缓存、网络软件栈、用户态缓冲区等位置。另外，现有的软件系统大多采用传统的中断机制，以响应用户请求，这种方式的时延一般在微秒甚至毫秒级，过于低效。

综上，简单地将现有的存储软件部署到持久性内存上，并不能充分发挥其硬件特性，甚至有可能导致软件错误、数据不可恢复等新的问题。因此，系统设计者必须充分了解持久性内存的性能和硬件特性，针对性地设计适合于持久性内存的存储系统软件。

3.3 持久性内存的存储系统构建方法

本节将分别从持久性内存的数据一致性管理机制、持久性内存文件系统、持久性内存的分布式存储系统构建3个方面阐述存储系统构建中的设计方法以及如何应对这3个方面的挑战。

3.3.1 新型数据一致性管理机制

为避免传统日志机制引入的额外开销，需要设计全新的面向非易失内存的数据管理方式。本节主要从软件和硬件的角度，分别阐述降低顺序性和一致性开销的优化策略。其中，顺序性开销是指处理器数据根据数据依赖关系有序地持久化到非易失内存中的开销，而持久性开销指的是数据从多级易失性处理器缓存替换到非易失内存过程中可能存在的冗余持久化开销。

(1)降低顺序性开销的方法

大量研究工作通过在处理器缓存中以硬件的方式提供顺序性的支持，从而降低软件显式顺序性的开销。微软研究院在处理器缓存中增加新的原语指令[15]，该方法可以将程序划分成多个执行单元，这种机制保证了不同执行单元之间依旧遵循持久化顺序约束，而每个执行单元内部可以对写操作进行重排序，从而提升性能。英特尔公司于2014年设计了新的扩展指令，其中，clwb指令既能避免持久化指令之间的依赖关系，又可以避免写回的缓存行数据失效，从而使得持久化的数据依旧供后续访问继续使用，减少缓存缺失操作带来的性能影响。当上层应用需要保证持久化操作的顺序时，它们可以通过内存屏障指令（例如mfence）控制持久化操作的顺序。

（2）降低持久性开销的方法

部分做法是设想处理器缓存的部分或所有层次采用非易失性存储器，从而缩短持久化路径，降低持久化开销。微软研究院提出了全系统持久化（whole system persistence，WSP）技术[16]，使所有处理器缓存均采用非易失存储器，并采用后备电源的方式保证在系统掉电后总线上的数据传输。在软件方面，清华大学设计了BPPM[17]，由于日志保证了已经提交的数据的持久性，因此在将数据写回数据区的过程中，数据无须立即持久化，只有当日志空间不足时，才将缓存在DRAM中的数据持久化到非易失内存中，从而减少了持久化带来的时延。

3.3.2 更精简的持久性内存文件系统

文件系统是操作系统中最基础的模块，也是存储系统中应用较广泛的抽象模式。它将设备存储空间以文件的形式组织为可索引的文件目录树，从而方便用户存取数据。为兼顾现有的应用程序，将非易失内存通过文件系统进行组织。一种便捷的方式是直接使用现有的外存文件系统管理持久性内存空间。例如，通过虚拟内存盘（RAMDISK）将非易失内存抽象成块设备，这样现有的外存文件系统（如EXT4、XFS、BtrFS等）均可直接部署在该块设备上。这种途径无须对文件系统做出任何修改，这使得传统文件系统可以快速获取大幅的性能提升。

然而，上述方法的缺陷是软件层次开销大，无法充分利用非易失内存的优势。近年来，已经有大量的工作专门针对持久性内存设计新的文件系统。本节将从移除DRAM缓存、构建用户态文件系统两个方面阐述对文件系统进行的相关优化。

（1）移除DRAM缓存

Linux内核中现有的文件系统模块均为外存设计，为了提升性能，VFS专门管理了一部分DRAM空间，用于缓存最近访问的文件数据。然而，NVM具有与DRAM接近的性能，因此，DRAM缓存不再具有缓存效果，相反地，还引入了额外的内存复制，严重影响性能。针对这个问题，EXT4、BtrFS等传统的文件系统均兼容了直接访问（direct access，DAX）模式。通过这种方法，应用程序可以直接访问非易失内存中存储的文件数据，而不需要将数据复制到DRAM缓存空间中。针对非易失内存重新设计的PMFS[18]、NOVA[19]、BPFS[15]等文件系统，则通过内存映射的方式绕开了文件系统页缓存，从而避免了数据的冗余复制。

（2）构建用户态文件系统

虽然大多数持久性内存文件系统均引入了DAX模式，以消除DRAM缓存带来的额外开销，但是将文件系统构建在内核态依旧无法避免现场切换以及VFS带来的开销。因此，一种可行的方案是将文件系统直接部署到用户态，例如Aerie[20]、Strata[21]等文件系统。它们将持久性内存空间直接映射到用户态，并通过一个用户库封装了文件系统访问接口，因此，应用程序可以直接在用户态访问文件数据，从而消除了操作系统引入的额外开销。

3.3.3 基于RDMA的持久性内存的分布式存储系统

为满足大规模数据处理对存储容量的需求，还需要将集群中各机器的持久性内存统一组织起来，构建大规模的分布式持久性内存存储系统。远程直接内存访问（remote direct memory access，RDMA）能够在远端处理器不参与的情况下直接读写远端内存，从而提供零复制的数据传输能力。迈洛斯公司最新发布的

ConnectX-6系列网卡已经支持200 Gbit/s的数据传输带宽和亚微级秒的传输时延。

持久性内存和RDMA分别在存储和网络上提供了极高的硬件性能。然而，现有的分布式软件系统设计复杂、层次冗余，引入了极高的软件开销[22]。清华大学于2017年提出的分布式持久性内存文件系统Octopus[23]正是为了解决这个问题，重新设计了文件系统软件栈。Octopus将各存储节点的NVM通过RDMA统一互连起来，构建成一个统一寻址的持久性共享内存池（如图3所示），通过这样的抽象，客户端可以直接通过RDMA网络读写内存池中的文件数据，极大精简了软件逻辑，降低了冗余复制。

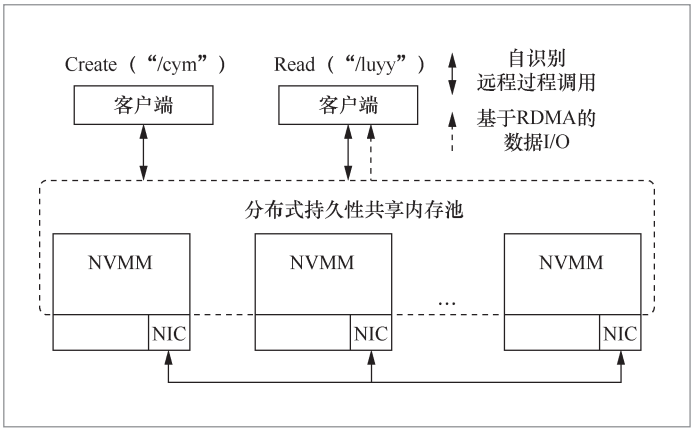


图 3 Octopus 文件系统架构

3.4 小结

非易失内存具有不同于现有任何存储介质的硬件属性，这为系统设计人员构建存储系统带来了一致性数据管理、操作系统架构、分布式软件设计等方面的挑战。现有的工作已经从持久性内存的空间管理、编程模型设计、索引结构、文件系统、分布式存储系统等方面展开了深入的研究，有效地解决了非易失内存中存在的若干问题。

4 未来存储系统发展的若干思考

高并发、低时延、细粒度等将是未来大规模应用对数据中心存储系统的主流访问特征，这对存储系统的任务调度、数据索引与管理、数据中心架构等方面都带来了极大的挑战。为应对这些问题，本文从异构系统下的存储计算融合架构以及新型数据中心架构角度展望了存储系统的发展趋势。

（1）存储计算融合的闪存存储架构

在本地存储中，闪存设备具有极高的内部并发能力，内部的传输带宽大于主机与设备之间的传输带宽。为了减少主机与设备之间的数据传输，一种思路是将主机端的部分计算卸载到数据所在的设备上，这样还可以达到充分利用设备内带宽资源的目的。在分布式存储中，通信、分布式协议存在较大开销，一种思路是在网络硬件上进行一些通用计算，从而实现系统功能优化，降低分布式协议开销。上述方法的核心是将计算处理单元部署到离数据存储更近的地方，缩短I/O处理路径，减少服务器带宽占用，即近数据处理。当前闪存固态盘架构中，设备内控制器具有计算能力，进行闪存控制、主机交互和FTL的运行。除此之外，设备内还可以增加现场可编程门阵列（field programmable gate array，FPGA）硬件来加速部件或通用处理单元，提供更强的计算处理能力。另外，以可编程网卡（programmable NIC）和可编程交换机（programmable switch）为代表的可编程网络硬件近年来也得到了飞速发展，为分布式存储的设计提供了新的机遇，为实现低时延的分布式系统提供了硬件支持。目前，开放通道闪存提供了一种软件与硬件

协同设计的思路，如何进一步结合可编程硬件和存储计算融合的思想，从构建大规模闪存存储系统的角度，实现存储计算功能的全局（存储级、节点级和网络级）合理分布，是值得关注和研究的问题。

（2）Rack-Scale的数据中心架构

传统数据中心是由单个的服务器节点组建起来的，每个服务器节点内部包含了外存设备、内存、CPU等硬件资源，这些服务器通过多层网络互连，构成大规模分布式集群。如果想对集群进行扩容，只能购置更多的服务器节点，必要时，还得额外购买机架和交换机。这种数据中心架构方式具有资源利用率低、数据中心部署灵活性差、难以扩展等问题。一种新的途径是将服务器内部的各种硬件资源拆分开，将不同类的硬件资源构建成硬件资源池，并通过高速网络进行互联。通过这种方式，数据中心的扩展不再以“服务器”为粒度，而是直接以机架为单位进行扩展，这种数据中心的构建模式叫作Rack-Scale架构，它具有部署容易、升级容易、资源管理更灵活等优点。因此，如何基于这种全新的数据中心架构设计存储系统，是未来考虑的重点。目前Rack-Scale的发展依旧处于初级阶段，其中最大的阻碍就是新一代网络互联系统还没有成熟。这是因为将内存和CPU拆分开之后，所有的内存访问都要通过网络传输，这给高速网络的设计带来了极大的挑战。

5 结束语

大数据存储系统日益难以满足全球快速增长的数据存储需求，“存储墙”问题凸显，大数据不止更“大”，还要更“快”。从传统磁盘到闪存固态盘，实现了从机械式到电子式的跨越，开放通道闪存设备则从软硬件协同设计的角度，提出了闪存存储系统构建的新思路。非易失内存作为新的存储层级，同时提供了内存级访问性能及持久性存储特性，针对非易失内存设计更快的存储系统，得到了研究人员广泛关注。本文从闪存存储、持久性内存存储两种存储系统构建方案出发，详细阐述了其各自面临的挑战以及相应的解决方案，最后，展望了未来数据中心及存储系统构建的若干发展趋势。

参考文献:

[1] SHARMA S. Trends in server efficiency and power usage in data centers[R]. Auckland: SPEC 2016 ASIA SUMMIT, 2016.

[2] AJOUX P, BRONSON N, KUMAR S, et al. Challenges to adopting stronger consistency at scale[C]//The 15th Workshop on Hot Topics in Operating Systems (HotOS XV), May 18-20, 2015, Switzerland. Berkeley: USENIX Association, 2015.

[3] AGRAWAL N, PRABHAKARAN V, WOBBER T, et al. Design tradeoffs for SSD performance[C]// USENIX Annual Technical Conference, June 22-27, 2008, Boston, USA. Berkeley: USENIX Association, 2008: 57.

[4] LU Y Y, SHU J W, ZHENG W. Extending the lifetime of flash-based storage through reducing write amplification from file systems[C]//The 11th USENIX Conference on File and Storage Technologies, February 12-15, San Jose, USA. Berkeley: USENIX Association, 2013: 257-270.

[5] LEE S, LIU M, JUN S, et al. Application-managed flash[C]//The 14th USENIX Conference on File and Storage Technologies, February 22-25, 2016, San Jose, USA. Berkeley: USENIX

Association, 2016: 339-353.

[6] OUYANG J, LIN S, JIANG S, et al. SDF: software-defined flash for web-scale internet storage systems[J]. ACM SIGPLAN Notices, 2014, 49(4): 471-484.

[7] ZHANG J, LU Y, SHU J, et al. FlashKV: accelerating KV performance with open-channel SSDs[J]. ACM Transactions on Embedded Computing Systems, 2017, 16(5s): 139.

[8] BJØRLING M, GONZÁLEZ J, BONNET P. LightNVM: the Linux open-channel {SSD} subsystem[C]//The 15th USENIX Conference on File and Storage Technologies, February 27-March 2, 2017, Santa Clara, USA. Berkeley: USENIX Association, 2017: 359-374.

[9] HUANG F B. Achieving both performance isolation and uniform lifetime for virtualized SSDs[C]//The 15th USENIX Conference on File and Storage Technologies, February 27-March 2, 2017, Santa Clara, USA. Berkeley: USENIX Association, 2017.

[10] ZHANG J, SHU J, LU Y. ParaFS: a log-structured file system to exploit the internal parallelism of flash devices[C]//2016 USENIX Annual Technical Conference, June 22-24, 2016, Denver, USA. Berkeley: USENIX Association,2016: 87-100.

[11] CARLSON M. Storage for hyperscalers[R]. Santa Clara: SNIA 2016, 2016.

[12] YANG J, PLASSON N, GILLIS G, et al. Don't stack your log on my log[C]// The 2nd Workshop on Interactions of NVM/Flash with Operating Systems and Workloads, October 5, 2014, Broomfield, USA. [S.l.:s.n.], 2014.

[13] IZRAELEVITZ J, YANG J, ZHANG L, et al. Basic performance measurements of the Intel Optane DC persistent memory module[J]. Computer Science, 2019, arXiv:1903.05714.

[14] SWANSON S. Redrawing the boundary between software and storage for fast non-volatile memories[R]. San Diego: University of California, 2015.

[15] CONDIT J, NIGHTINGALE E B, FROST C, et al. Better I/O through byte-addressable, persistent memory[C]// The ACM SIGOPS 22nd Symposium on Operating Systems Principles, October 11-14, 2009, Big Sky, USA. New York: ACM Press, 2009: 133-146.

[16] NARAYANAN D, HODSON O. Whole-system persistence[J]. ACM SIGARCH Computer Architecture News, 2012, 40(1): 401-410.

[17] LU Y Y, SHU J W, SUN L. Blurred persistence in transactional persistent memory[C]//The 31st Symposium on Mass Storage Systems and Technologies (MSST), May 30-June 5, 2015, Santa Clara, USA. Piscataway: IEEE Press, 2015: 1-13.

[18] DULLOOR S R, KUMAR S, KESHAVAMURTHY A, et al. System software for persistent memory[C]//The 9th European Conference on Computer Systems, April 14-16, 2014, Amsterdam, the Netherlands. New York: ACM Press, 2014: 15.

[19] XU J, SWANSON S. NOVA: a log-structured file system for hybrid volatile/non-volatile main memories[C]//The 14th USENIX Conference on File and Storage Technologies, February 22-25, 2016, Santa Clara, USA. Berkeley: USENIX Association, 2016: 323-338.

[20] VOLOS H, NALLI S, PANNEERSELVAM S, et al. Aerie: flexible file-system interfaces to storage-class memory[C]// The 9th European Conference on Computer Systems, April 14-16, 2014, Amsterdam, the Netherlands. New York: ACM Press, 2014: 14.

[21] KWON Y, FINGLER H, HUNT T, et al. Strata: a cross media file system[C]// The 26th Symposium on Operating Systems Principles, October 28-31, 2017, Shanghai, China. New York: ACM Press, 2017: 460-477.

[22] 陈游旻, 陆游游, 罗圣美, 等. 基于RDMA的分布式存储系统研究综述[J]. 计算机研究与发展, 2019, 56(2): 227-239.
CHEN Y M, LU Y Y, LUO S M, et al. Survey on RDMA-based distributed storage systems[J]. Journal of Computer Research and Development, 2019, 56(2): 227-239.

[23] LU Y Y, SHU J W, CHEN Y M, et al. Octopus: an RDMA-enabled distributed persistent memory file system[C]//2017 USENIX Annual Technical Conference (USENIX ATC 17), July 12-14, 2017, Santa Clara, USA. Berkeley: USENIX Association, 2017: 773-785.

作者简介

陈游旻(1993-), 男, 清华大学计算机科学与技术系博士生, 主要研究方向为文件系统、分布式系统。

李飞(1993-), 男, 清华大学计算机科学与技术系硕士生, 主要研究方向为闪存存储系统。

舒继武(1968-), 男, 博士, 清华大学计算机科学与技术系教授, 教育部长江学者特聘教授, IEEE Fellow, 博士生导师。近年来主要从事基于非易失存储器件的新型存储系统与技术、基于Flash器件的固态存储系统与技术、网络(云/大数据)存储系统与关键技术、数据存储可靠性等方面的研究工作。

收稿日期: 2019-05-07
基金项目: 国家重点研发计划基金资助项目(No.2018YFB1003301)
Foundation Item: National Key Research and Development Program of China(No.2018YFB1003301)