2018年IEEE第24届平行和分布式系统国际会议（ICPADS）。

虚拟化和非虚拟化环境中的3D XPoint SSD的性能分析

张家宸，李鹏，刘波，Trent G. Marbach，刘晓光*，王刚*。
南开大学计算机学院南开-百度联合实验室，天津，中国
{jczhang, lipeng, liubo, trent.marbach, liuxg, wgzwp}@nbjl.nankai.edu.cn

***摘要*--英特尔的Optane SSD最近作为基于3D XPoint的商业设备的先驱进入了市场。与传统的SSD相比，它们具有更低的延迟（约14μs）和更好的并行特性，因此，在商业环境中，它们将取代NAND闪存SSD。为了更好地满足云计算和企业数据中心的高性能存储需求，有必要了解新设备在虚拟化云环境和传统非虚拟化环境中的性能特征。在本文中，我们介绍了基于大量实验的Optane SSD的分析。我们使用几个微观的测试来获得Optane的基本性能指标的知识。我们还讨论了最先进的存储堆栈对Optane SSD性能的影响。通过分析测试结果，我们为使用Optane SSD的存储I/O应用提供了配置建议。最后，我们通过运行基于MySQL数据库的实验来评估Optane SDD的实际性能。所有的实验都是在非虚拟化和虚拟化环境（Linux和QEMU）下进行的，并对Optane SSD和基于SATA NAND闪存的SSD进行了比较研究。**

***Index Terms*-3D XPoint, solid-state drive (SSD), non-volatile memory (NVM), virtualization, performance analysis**

978-1-5386-7308-9/18/$31.00 ©2018 IEE doi 10.1109/icpads.2018.00018

I. 简介

目前，存储硬件正在迅速发展。基于NAND闪存的固态硬盘（SSD）由于比旋转式硬盘（HDD）更有优势，在企业存储系统中作为第二存储介质已经非常流行。新兴的非易失性存储器技术（NVMs），如PCM、STT-RAM、ReRAM和3D XPoint，也在积极开发中。与NAND闪存设备相比，NVMs具有更低的延迟、更高的带宽和更长的使用寿命。

如图1所示，基于NAND闪存的SSD通常通过SATA或PCIe连接，而基于NVM的设备可以通过DIMM或PCIe连接，因此与SSD和DRAM设备竞争。已经有很多关于NVM的研究[1]。然而，大多数研究都是基于NVM仿真器的，因为基于NVM的设备以前没有商业化。最近，基于3D XPoint的英特尔Optane SSD（Optane）被发布到市场上。

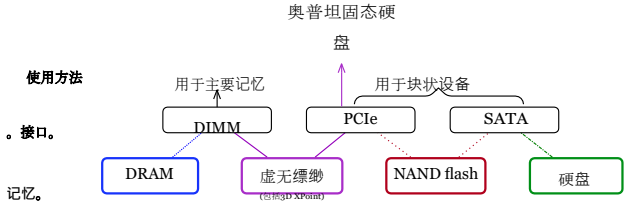


图1：记忆和它们的使用模式。

ket。这些设备通过PCIe接口连接，基于NVMe协议规范[2]。虽然PCIe接口不是革命性的，但底层的3D XPoint内存与NAND闪存有许多不同的特点。因此，我们对Optane在以下方面的表现感兴趣。

1) 目前的现代操作系统（OS）存储堆栈，包括块I/O子系统和文件系统等层，最初是为旋转磁盘优化的，后来又为基于NAND闪存的SSD优化[3]。这些技术对于Optane来说是否足够高效？
2) 虚拟化执行环境是云服务的后端，已经成为数据中心不可或缺的东西。然而，由于额外的虚拟化层和延长的I/O路径，它们的存储堆栈更加复杂。Optane设备在虚拟化环境中的表现如何？
3) 存储I/O密集型应用，如关系型数据库管理系统（DBMS），主要是在存储过程中出现瓶颈。此外，以前的配置设置已经通过广泛的研究进行了优化，以便在传统设备上运行。在使用Optane时，应用程序的性能是如何提高的，以前的配置在使用Optane时是否仍然可行？

为了回答这些问题，我们在英特尔Optane SSD 900P上进行了广泛的实验，在Linux这个非虚拟化的物理环境（PE）和QEMU这个虚拟化的执行环境（VE）中。为了比较，我们还在基于NAND闪存的SATA SSD（NAND）上进行了每个实验。对于性能实验，我们使用了微观测试和真实世界的应用MySQL OLTP基准。

本文的其余部分将按以下步骤进行。第二节讨论了存储堆栈，第三节对基本指标进行了测量。我们在第四节中给出了使用Optane的建议，并在第五节中评估了Optane配备的MySQL。最后，第六节介绍了相关工作，第七节总结了本文的工作。

II. 系统软件的影响

A. 仓储堆栈

由于系统扩展所需的高灵活性，计算机系统，如存储系统，通常使用多层次的间接性。然而，通信

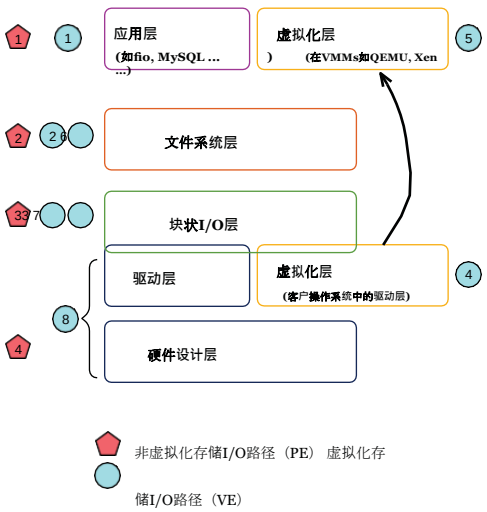
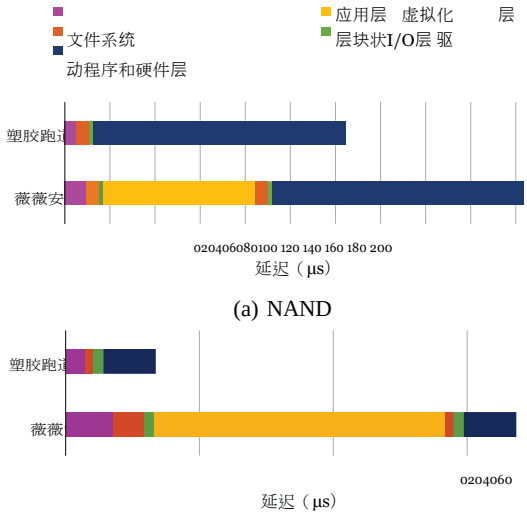


图2：PE或VE存储堆栈中的I/O路径。



(a) NAND

(b) 芯片

图3：VE和PE存储栈中4KB随机读取I/O请求延迟的构成。

在不同的层之间，将不可避免地导致额外的性能开销[4]，从而增加请求延迟。

图2显示了PE和VE的典型存储堆栈，分别用五边形和圆圈符号标注。图中有序的数字表示一个I/O请求所经过的层。例如，在PE中，一个读请求将从顶部的应用层传到底部的硬件设备层，编号为1
4。在VE中，情况变得更加复杂，虚拟机（VM）在基于主机操作系统的称为虚拟机监控器（VMM）的应用程序中运行。

由虚拟机中的应用程序发出的请求将首先经过虚拟机存储栈的各层，然后再经过主机服务器的存储栈，编号为1
8。这两个
额外的虚拟化层（编号为4和5）作为
是虚拟机和主机服务器之间的桥梁。

存储堆栈的整体延迟包括硬件设备层和驱动层产生的与硬件相关的延迟，以及堆栈的其他部分产生的与软件相关的延迟。

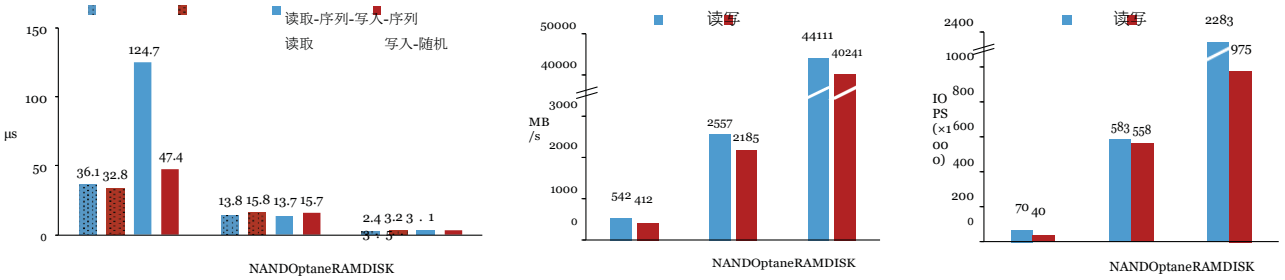
在过去，数据中心主要依靠PEs。再加上当时的磁盘速度，这意味着硬件延迟在很大程度上主导了总请求延迟[5]。然而，最近情况在两个方面变得更加复杂：（1）VE越来越流行，随着层数的增加而延长了软件延迟，并因此减少了硬件延迟的比例。(2)
更高速度的设备（如Optane）已经出现，这大大降低了硬件延迟。为了清楚地了解如何优化总延迟，我们测量了配备NAND和Optane设备的PE和VE中不同层所产生的延迟细分，这将在下一节中详细说明。

B. 延迟细分

在本节中，我们评估了在PE和VE中使用NAND和Optane时的延迟故障。我们使用
与第三节中的实验设置相同。我们还测量了随机读取请求，因为它们在现实世界的工作负载中是很常见的，借助于`fio`[6]和`blktrace`[7]。每层的延迟都是直接测量或计算出来的。结果显示在图3中，各层的颜色与图2中的颜色相对应。我们做了几个观察。

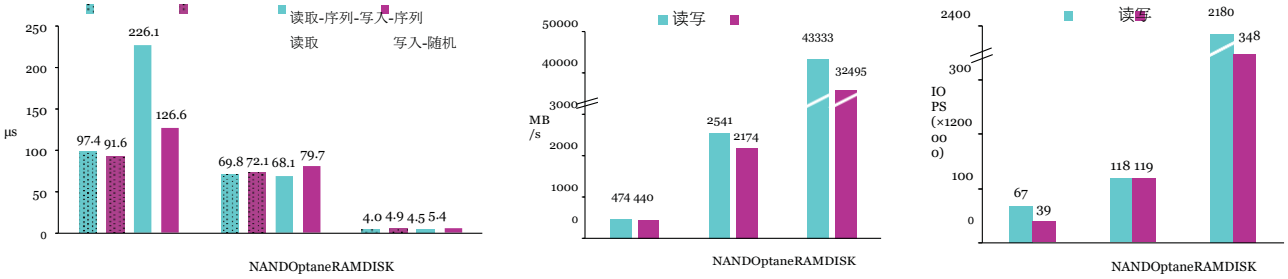
1) 正如预期的那样，使用Optane降低了两种环境下的延迟。具体来说，由于使用Optane时的硬件延迟为7.9μs，比NAND的延迟快103.8μs，因此在PE和VE中，总体延迟将分别减少67.0%和89.1%。
2) VE中的延迟比PE中的延迟要高，两者都是如此。由于VE的I/O路径的扩展，增加了软件延迟。具体来说，很大一部分的软件延迟是由虚拟化层造成的。因此，在使用NAND和Optane时，VE的软件延迟分别占总延迟的45.1%和88.3%。因此，使用VE的NAND设备的整体延迟（203.7μs）是使用PE时（67.3μs）的3倍，而使用Optane时，这一因素增加到10倍（VE中124.5μs，PE中13.5μs）。
3) 在PE中，NAND的软件延迟可以忽略不计。与其整体延迟相比，只有10.3%的比例。然而，由于Optane实现了较低的硬件延迟，这一比例对于Optane来说增加到41.6%。

在这些事实的指导下，我们提出，减少整个请求的延迟可以通过以下方式最好地实现。(1)
采用新设备，这有助于减少硬件延迟。(2)
对VE中的虚拟化层进行优化，这将大大减少VE中的软件延迟。(3) 对传统操作系统存储的优化



(a) 延迟 (b) 带宽 (c) IOPS

图4：NAND、Optane和RAMDISK在一般Linux环境下的微观测试（PE）。



(a) 延迟 (b) 带宽 (c) IOPS

图5：NAND、Optane和RAMDISK在虚拟化的QEMU环境（VE）中的微型测试。

PE中的堆栈，这将减少PE和VE中的请求延迟。

III. 基本指标

在本节中，我们将展示Optane在以下方面的表现通过与PE和VE中的NAND和RAMDISK两种设备进行比较，得出了延迟、IOPS和带宽等基本指标。这些设备的详细信息列于表I。Optane是一个使用3D XPoint技术的PCI-连接设备，而NAND是一个基于NAND闪存的传统SATA连接设备，RAMDISK是一个使用DRAM的模拟设备。

我们在第三节C中进一步探讨了延迟和IOPS之间的性能曲线，并提出了一个总结。

表一：硬件设备。

	NANDOptaneRAMDISK		
设备 型号Intel	S3510IntelOptane 900PMicronDDR4		
接口SATA	3.0PCIe3.0×4DIMM		
存储介质NAND	flash3D XPointDRAM		
容量480	GB480	GB64GB	
每GB0.	681.	1613.13美元	

A. 实验设置

我们使用`fio`来进行测试，通过生成适当数量的读或写线程和适当的I/O请求大小，这将在第三节B中详细说明。

所有的实验都是在X86服务器上进行的，其配置信息列于表二。对于PE，所有

表二：服务器配置。

CPUIntel Xeon E5-2609 1.70 GHz ×2	
CPU核数	8 2
处理器高速缓存	32 KB L1i, 32 KB L1d, 256 KB L2, 20 MB L3
DRAM128 GB	
OSRHEL 7.0，内核版本4.14（在虚拟机中相同）	
VMMQEMU 2.10	
文件系统	XFS

设备采用XFS文件系统进行格式化。对于VE，我们在Optane和NAND设备上创建原始格式的镜像文件，作为虚拟机存储的后端。RAMDISK是由客户操作系统在虚拟机中直接创建的，因为内存虚拟化比存储I/O虚拟化更有效（由于延长了VE的I/O路径）。请注意，在VE中，RAMDISK的额外开销主要是由内存虚拟化而不是额外的存储堆栈层造成的。我们还在VE中用XFS格式化这些设备。

在所有的实验中，每个结果都是在30s的执行过程中的平均值，存储在硬盘中的数据大小保持在20GB（最初随机设置）。

B. 延迟、带宽和IOPS

a) I/O 延迟。为了测量一个I/O请求的平均延迟，我们在单线程模式下运行`fio`，每次产生4KB的请求。

PE和VE延时实验的结果显示分别为图4（a）和图5（a）。我们使用*Read-seq*和*写入-seq*表示顺序读和写测试，以及

*Read-rand*和*Write-rand*分别表示随机读取和写入测试。在PE中，正如预期的那样，Optane实现了比NAND更低的延迟，在所有四种请求类型中平均减少了67.4%。此外，Optane的延迟在顺序和随机请求之间更加平衡，最多只有12.7%的差异，而NAND请求之间的差异为2到3倍[1]。在VE中，所有情况下的延迟都会增加，然而，Optane的性能受额外的虚拟化层的影响最大。例如，平均来说，Optane在VE中的随机读取延迟是它在VE中的5倍。

PE，而对于NAND，乘法系数下降到2次。

在所有情况下，RAMDISK都是最快的设备，这是因为使用DRAM获得的内存速度更快。然而，Optane平均只比RAMDISK慢5倍，而NAND则比RAMDISK慢10至40倍。RAMDISK也实现了相对平衡的速度（例如，在PE中约3µs，在VE中约5µs），显示了Optane和RAMDISK之间的相似性。

*b) 传输带宽：*我们通过在多个线程上向设备发送多个请求单元（128KB）来计算I/O吞吐量（传输带宽）的上界值。

图4（b）和图5（b）显示了PE和VE中的带宽。可以看出，Optane的带宽超过2GB/s，大约是NAND带宽的5倍。由于Optane是通过PCIe 3.0 4接口连接的，其理论带宽为3.94 × GB/s，而NAND是通过SATA 3.0连接的，其理论带宽为600 MB/s，接口对带宽的限制并不强烈。在VE中，较大的128KB单元的传输时间较长，加上多线程的使用，隐藏了额外的虚拟化延迟。

尽管RAMDISK在每种内存类型中拥有最好的带宽，但与NAND相比，使用Optane的性能提升仍然是有意义的。Optane的更高带宽表明它可以被用作日常工作负载中的持久性文件缓存，如备份系统。此外，更高的带宽可能会改变一些存储服务的优化策略，如透明数据压缩、重复数据删除和清除编码。我们在第四章C节通过对透明压缩的案例研究进行了更详细的分析。

*c) 最大的IOPS。*为了测量每个设备的最大IOPS，我们在多线程模式（64 个线程）下运行`fio`，它产生随机读写请求，I/O大小为4KB。我们通过平均每秒钟的成功请求数来计算IOPS。使用小的单位大小可以节省I/O带宽；因此我们可以期望在达到硬件传输带宽之前达到最高的IOPS。

图4(c)显示，Optane在PE中的效果明显优于NAND，比NAND高一个数量级。

[1]从理论上讲，由于NAND闪存的写入放大系数，NAND的 写入操作应该比读取操作更昂贵[8]，我们认为这里的随机写入比随机读取快的原因是由于我们设备中的写入缓冲[9]。

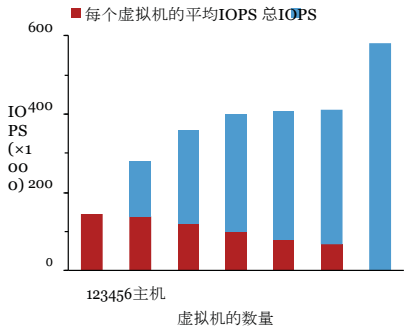


图6：基于一个Optane的多个虚拟机的读取IOPS。

IOPS（从8倍增加到11倍）。对于一个给定的参数设置，IOPS的理论上限是由其相应的带宽除以4096（I/O大小）得出的。NAND获得了51.7%和38.8%的读写上限，Optane获得了91.2%和102.15%[2]，而RAMDISK获得了20.7%和9.7%。从这些结果可以看出，大的线程和请求数对NAND和RAMDISK设备有负面影响，但似乎并不影响Optane设备。这些结果表明，Optane在并行设置中运行良好。

此外，Optane的读和写操作之间只有很小的差距，但NAND的差距很大。这表明，Optane自然适合不同的读写情况，而NAND由于NAND闪存的写放大系数，在写密集型系统中表现不佳。由于内存总线的高度并行性，RAMDISK的IOPS表现最好，比Optane高2到4倍。

在VE中，如图5（c）所示，Optane只获得了大约12万的IOPSs，是PE中的五分之一。相比之下，NAND获得的IOPS与PE中的IOPS几乎相同。这表明在VE中，Optane的并行性仍然没有得到充分的利用。

为了探索如何在VE中充分利用Optane，我们部署了多个虚拟机，这些虚拟机使用Optane作为其存储后端。我们在这些虚拟机中同时运行`fio`基准测试，并测试所有虚拟机的累积IOPS。结果如图6所示，其中蓝色条表示总IOPS，红色条表示平均IOPS。我们观察到，尽管现代VMM在单个虚拟机的并行性方面为Optane提供了足够的支持，但在部署多个虚拟机时，Optane仍然可以在VE中发挥一定作用。随着虚拟机数量的增加，IOPS呈亚线性改善，直到虚拟机数量为4。当使用6 个虚拟机时，IOPS达到饱和，约为41万IOPS，这与PE中的IOPS（58万）相当。这意味着，如果利用得当，Optane可以成为云数据中心内一个有前途的解决方案。

[2]这个数字之所以略大于100%，是因为我们使用的理论上限也是在上一节中测量的，测量误差是不可避免的。

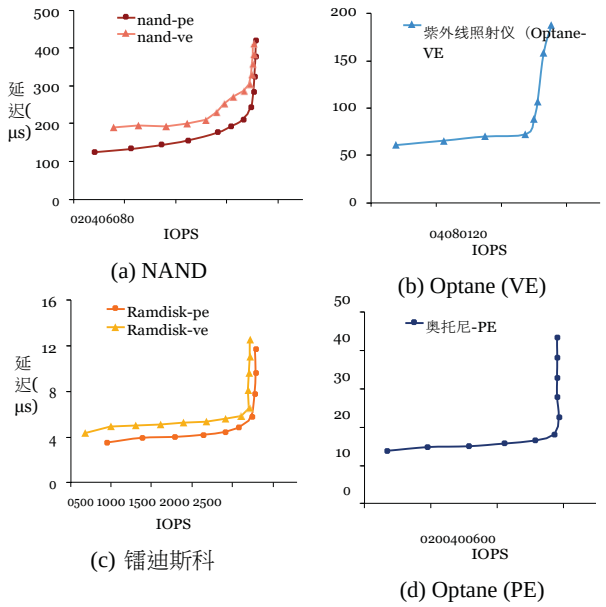


图7：IOPS和延迟之间的关系。

C.性能曲线（在延时和IOPS之间）。

为了更好地理解这些设备的并行性质，我们还测试了随机读取请求的IOPS和延迟之间的关系，并绘制了延迟曲线，见图7。

随着IOPS率的增加，设备的总需求量更大。随着IOPS的增加，传统设备的平均延迟逐渐变差，而Optane和RAMDISK设备对这种需求的适应性更强。一旦IOPS达到上限，所有设备的延迟都会迅速增加 [10] [11]。

从图7（a）、（c）和（d）中可以看出，当IOPS率达到95%的上限时，NAND、RAMDISK和Optane的延迟分别比其基本水平增加了80%、54%和25%。这表明Optane在内部花了更少的时间来处理数据。因此，Optane的延迟可以在高并发工作负载中保持最佳状态。此外，当Optane达到其IOPS极限（约58万）时，相应的吞吐量约为2.2 GB/s，这非常接近其传输带宽（2.5 GB/s）。相反，NAND在达到其70k的IOPS上限时，得到了280MB的吞吐量，这比其带宽（542MB/s）小得多。因此，我们得出结论，Optane具有更好的并行性和可扩展性，在高并发工作负载方面表现更好。图7（a）、（b）和（c）显示了VE中NAND、Optane和RAMDISK的结果。NAND和RAMDISK的性能只比PE中的性能差一点，而Optane的性能在VE中则下降了很多。然而，随着IOPS率的增加，Optane的延迟仍然比较稳定。当IOPS率达到95%的上限时，NAND、Optane和RAMDISK的延时分别增加了158%、50%和35%。

通过这些测量，我们相信尽管Optane每GB的价格是NAND设备的两倍，但它仍然是一个

对于现代数据中心来说，Optane比NAND更有竞争力的选择。另一方面，与NAND相比，Optane的性能在VE中下降得更多。这表明，为了在云数据中心更好地利用Optane，应该对调度和配置问题进行进一步研究。然而，如上所述，由于一个虚拟机无法充分利用Optane的并行性，目前在云数据中心中，为多个虚拟机使用一个Optane设备更为合适。

IV. 对存储优化的影响

A.文件缓存

文件缓存被用来弥补存储设备和主存储器之间的性能差距。在I/O密集型应用的存储模块中，使用DRAM作为文件缓存是一种常见的优化。当使用文件缓存时，平均读取I/O延迟可以表示如下。

$$延迟 = t_{I/O} \times (1 - H) + t_{load} \times H \quad (1)$$

其中H表示缓存命中率，$t_{I/O}$ 表示平均从磁盘读取数据的延迟，t_{load} 表示从文件缓存中获取数据的时间。一般来说，$t_{load} \times H$一词可以省略，因为与$t_{I/O}$相比，t_{load}非常小。使用Optane时，文件缓存仍然是必要的，但不那么重要，因为它的速度（$t_{I/O}$）更接近DRAM（t_{load}）。此外，由于Optane比HDD快几个数量级，Optane本身可以作为HDD和主存储器之间的缓存[12]。

B.I/O颗粒度

在DBMS和键值存储等I/O密集型应用中，数据通常由固定长度的连续块组织，我们称之为页。一个页面是被获取或存储的最小的数据单位。

传统上，当使用HDD时，大的寻道时间t_{seek}主导了请求的响应时间，这导致了大的页面大小的选择。例如，ZFS[13]默认使用128KB。而作为一个常见的经验，当使用更快的设备时像SSD，较小的页面大小将是最佳的[14] [15]。然而，当设备从HDD和低端SSD切换到Optane时，这种经验可能是错误的，我们的目标是通过复杂的数学分析来探索内在原因。

一个应用程序使用3个步骤来提供查询请求。(1)解释请求以获得将被传输的相应页面，(2)在应用层和存储栈的底层软件层之间交换页面，以及(3)在底层软件层和存储设备之间交换数据。我们用T_{app}、T_{stk}和T_{dev}分别表示上述三个步骤所对应的时间成本。因此，一个请求的总时间成本可以描述为

$$t = t_{app} + t_{stk} + t_{dev}。 \quad (2)$$

请注意，在底层存储栈中，数据也是以页为单位组织的。我们用d_a和d_s分别表示应用层和底层存储栈层的页面大小。实际上，出于性能的考虑，d_a 是一个

d的倍数s，表明请求一个应用程序页面涉及d_a底层存储栈页面。此外，一个总的d_a的d_s如果要求的I/O大小为d，则将传输d个应用页，因此，我们有

$$T_{app} = t_{app} \times \frac{d}{d_a}, \quad (3)$$

$$T_{stk} = t_{stk} \times \frac{d}{d_a}, \quad (4)$$

其中t_{app}和t_{stk}表示一个页面在应用层和底层软件存储栈的时间成本。设b为存储设备的带宽，t_{seek}为存储设备的寻道时间，其值等于随机延迟和顺序延迟之差。因此，对于（2）中的第三项，我们有

$$T_{奉献} = \frac{d}{d_a} \times (t_{寻找} + \frac{d_a}{b}) 。 \quad (5)$$

将(3)、(4)和(5)应用于(2)，我们可以得到一个I/O大小为d的请求r_i的时间成本如下：

$$T = (t_{app} + t_{stk}) \times \frac{d_a}{d_s} + t_{seek} + \frac{d_a}{b} \times \frac{d_i}{d_a} \quad (6)$$

应用程序通常有两种类型的请求，可以称为点请求和范围请求。点请求只涉及比页面大小小得多的数据量，因此$\frac{d_i}{d_a}$将是1。范围请求涉及大量的数据，涉及许多页，因此$\frac{d_i}{d_a}$大约等于$\frac{d_i}{d_a}$。我们假设一个工作负载涉及N个查询请求，其中包括m个点请求和n个范围请求，由r_i（$1 \le i \le N, N = m + n$）索引，其请求的I/O大小由$D = d_1, d_2, \ldots, d_m, d_{m+1}, \ldots, d_N$表示。因此，给定工作负载的总时间成本可以表示为

$$\begin{aligned} T_D &= \sum_{i=1}^{m} T_i + \sum_{i=m+1}^{N} T_i \\ &= \sum_{i=1}^{m} (t_{app} + t_{stk} \frac{d_a}{d_s} + t_{seek} + \frac{d_a}{b}) \\ &\quad + \sum_{i=m+1}^{N} (t_{应用程} + t \frac{d_a}{ÅÅÅ} + t_{寻求} + \frac{d_a}{b}) \frac{d_i}{d_a} 。 \end{aligned} \quad (7)$$

让$\bar{d}$为范围请求的平均I/O大小。那么我们有

$$\begin{aligned} T_{Dappstk} &= m(t_{ds} + t \frac{d_a}{} + t_{seek} + \frac{d_a}{b}) \\ &\quad + n(t_{app} + t_{stk} \frac{d_a}{d_s} + t_{seek} + \frac{d_a}{b}) \frac{\bar{d}}{d_a} \end{aligned} \quad (8)$$

在（8）中，我们找到了T_D的导数的页面大小（d_a）。

$$\frac{\partial T_D}{\partial d_a} = m(\frac{t_{stk}}{d_s} + \frac{1}{b}) - \frac{n(t_{app}\bar{d} + t_{seek}\bar{d})}{d_a^2} \quad (9)$$

and there will be a extreme point of d_a

$$d_a = \sqrt{\frac{n(t_{app} + t_{seek})\bar{d}}{m(t_{stk}/d_s + 1/b)}} 。 \quad (10)$$

公式10提供了一种方法来确定理想的页面大小，这取决于应用程序的背景。当切换到Optane等高端固态硬盘时，硬件延迟t_{seek}和$1/b$已经足够小了，可以与软件延迟相媲美。因此，硬件延迟不再占主导地位。

公式10，软件因素t_{app}和t_{stk}（如应用延迟、应用工作负载、虚拟化延迟和操作系统I/O子系统延迟）将主导页面的最佳选择。

方程（10）中的尺寸。由于软件因素在执行环境和工作负载之间有很大不同，不仅倾向于为更快的设备选择更小的页面大小的经验变得无效，最佳选择也将更容易改变。因此，在使用Optane时，我们不应该只根据它的低硬件延迟来选择小的页面大小，需要更多与工作负载相关的分析和测试。

C. 面向存储的计算

面向存储的计算任务，如透明压缩、重复数据删除和擦除编码，被纳入一些I/O密集型应用中，以实现空间的高效性、高性价比。

性能和可靠性[16] [17] [18]。这些任务的工作与传统的较慢的存储设备相协调。然而。

当使用像Optane这样的新设备时，一些优势可能会消失，甚至可能导致性能下降。我们以透明数据压缩为例。

数据压缩已被广泛用于I/O密集型应用，如ZFS[13]、NTFS[19]和MySQL[20]，因为它提供了以下好处。(1）所需的存储空间更少，（2）获取数据的I/O（即磁盘和网络）带宽更低，以及（3）缓存命中率更高。

表三：I/O带宽和数据压缩吞吐量

(MB/s)

。

I/O		设备读与写
NAND542	412	
Optane2557	2185	
		算法解码Encoding
LZ42013	356	
Snappy915	269	
zlib defl	ate133	23

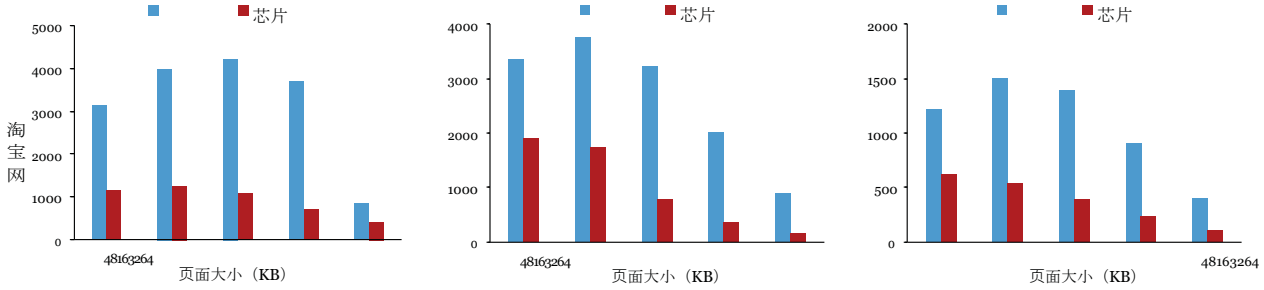
我们列出了几家公司的I/O带宽和编码吞吐量。表三中流行的无损压缩算法。I/O带宽取自第三节B，而编码吞吐量是由`lzbench`[21]测试的。压缩算法的测试是以块的颗粒度（128 KB）。传统上，在使用硬盘或NAND设备时。

解码比I/O读取速度快得多，这可以消除I/O流量并提高缓存容量，而编码不会因为广泛使用而使I/O写入出现瓶颈。

写缓冲。当涉及到Optane时，编码和解码都比I/O操作慢，这表明

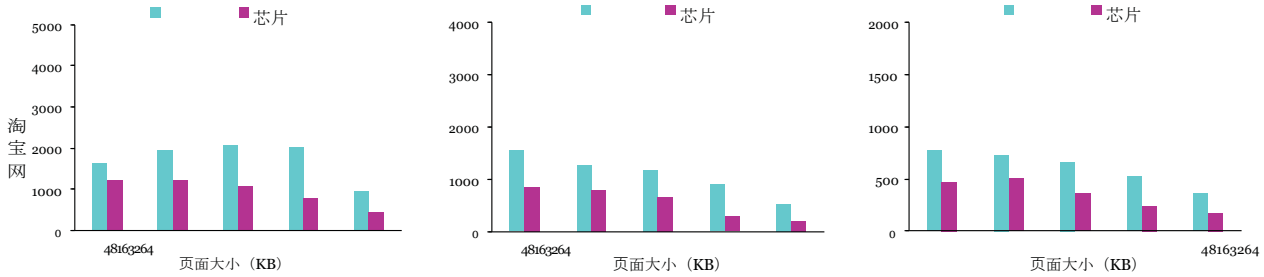
使用数据压缩可能导致性能不佳。

我们考虑了两种典型的应用场景，在这些场景中，使用Optane时，压缩可能不会对系统性能产生好处。首先，压缩数据可以提高缓存命中率



(a) 只读 (b) 只写的 (c) 混合读和写

图8：在PE中使用不同页面大小的MySQL OLTP性能。



(a) 只读 (b) 只写的 (c) 混合读和写

图9：在VE中使用不同页面大小的MySQL OLTP性能。

因为更多的数据可以在压缩后被缓存起来。然而，对于高速设备来说，解压缓存数据时可能会产生额外的延迟，这就抵消了高缓存命中率带来的好处。其次，透明压缩在传统上被用于系统中，以减少I/O带宽的数量，因为I/O延迟在总请求延迟中占主导地位，而不是计算。这个假设对于高速设备来说是不成立的，因为那里的解压缩成本比例相当大。因此，我们的结论是，在设计或配置配备Optane的应用程序时，数据压缩是存储空间节省和性能之间的权衡。

V. 数据库中的测试（mysql）。

很明显，直接转向Optane可以提高任何工作负载的性能。然而，在本节中，我们将重点关注Optane在不同配置的真实世界应用中的表现。我们选择了MySQL[20]，一个广泛使用的关系型数据库管理系统来进行测试。这些测试是根据以前的分析和结论来设计的。

我们在MySQL服务器5.7版本上运行由基准工具Sysbench[22]生成的MySQL在线事务处理（OLTP）测试。为了更好地观察，我们还做了一些配置上的决定。我们使用默认的存储引擎InnoDB，并启用直接I/O，因此操作系统页面缓存的影响被消除了。我们默认生成20GB的数据，使用32MB的文件缓存（在MySQL中称为缓冲池）。

InnoDB）和16 个客户线程连接到MySQL。在做混合读写测试时，读写请求的比例为7 : 2，这与Sysbench OLTP基准的默认设置一致。其他设置与第三部分相同。性能是以每秒交易量（TPS，越高越好）来衡量的。

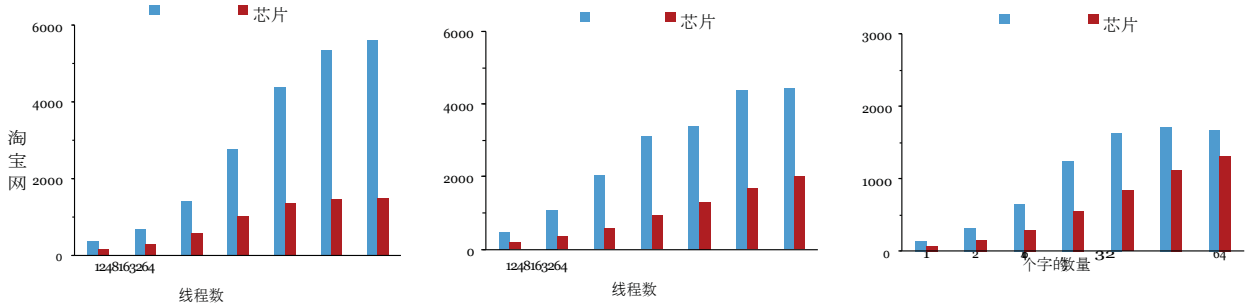
A. 页面大小

正如第IV-B节所解释的，当从NAND转换到Optane时，早期的经验[14][15]建议更快的设备采用更小的页面大小，这一点不再有效。这一结论在本实验中得到了验证。

在MySQL InnoDB中，所有的数据都是由被称为页的固定大小的块组织的，而页也是最小的I/O单元。因此，当页面大小在4KB和64KB之间变化时，我们测试OLTP性能。我们在图8和图9中显示了结果。在所有情况下，Optane仍然比NAND好得多，但是在VE中，Optane的TPS下降得比NAND多，这是由于大量的虚拟化开销造成的。在使用Optane时，我们更关注最佳页面大小。

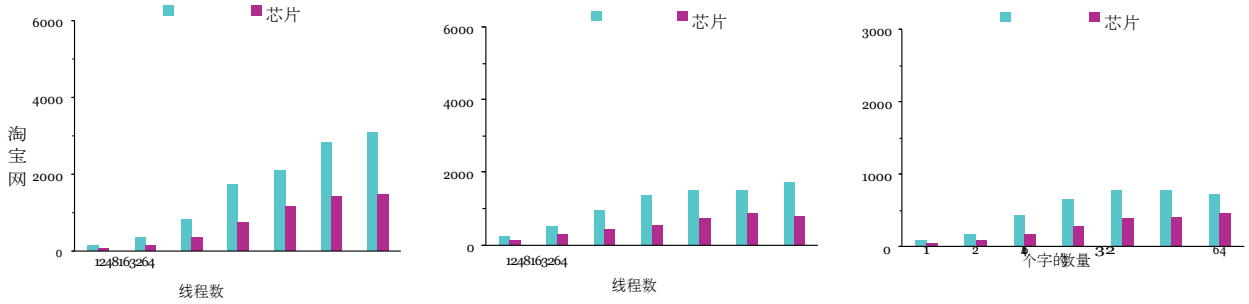
表四：最佳MySQL页面大小（KB）。

器材	阅读	混合R&W	撰写
芯片	16	8	8
Optane (VE)	16	4	4
NAND	8	4	4
NAND			(VE)884



(a) 只读　(b) 只写的　(c) 混合读和写

图10：在PE中使用不同线程数的MySQL OLTP性能。



(a) 只读　(b) 只写的　(c) 混合读和写

图11：在VE中使用不同线程数的MySQL OLTP性能。

我们在表四中列出了最佳页面大小。例如，在表四中，在两种环境中，只读情况的最佳页面大小是Optane的16KB和NAND的8KB，而在VE中，Optane的最佳页面比PE中的小。由于Optane优于NAND设备，而在PE中的Optane优于在VE中的Optane，这两个例子的结果都违背了早期的经验。因此，在配置配备Optane的MySQL时，需要根据实际工作负载进行测试或详细分析，以确定最佳页面大小。

B. 可扩展性

一个常见的情况是，一个MySQL服务器同时连接到多个客户端。在这种情况下，请求被同时提交给InnoDB存储引擎。因此，我们通过使用Sysbench提供的多线程基准来模拟这种情况。在PE和VE中的结果分别如图10和图11所示。正如预期的那样，使用Optane可以在所有情况下获得更好的TPS。如图10(a)所示，当只有一个线程连接时，Optane的TPS只比EPS高2.6。

NAND。当线程数为64时，Optane的TPS是比NAND高3.8。对于图10(b)和(c)中的只写情况和混合读写情况，当线程数小于8时，Optane获得了更好的可扩展性。

我们也相信NAND的良好可扩展性是由于我们的NAND硬件中的写缓冲，正如在第二章中解释的那样。

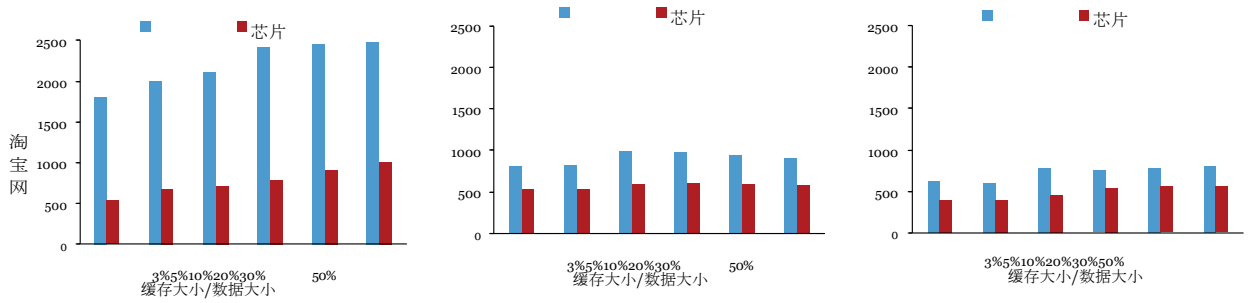
第三阶段。因此，我们得出结论，Optane在MySQL工作负载中具有更好的可扩展性，特别是对于读取工作负载。

与之前的测试类似，由于虚拟化层的大量开销，Optane在VE中的表现要差很多。如第三节所述，虽然一个虚拟机不能充分利用Optane的并行性，但我们可以使用多个虚拟机，并在其上运行MySQL，以获得整体上更高的可扩展性。这种情况在云计算服务中也比较常见，如数据库即服务（DBaaS）。

C. 缓存大小

如前所述，文件缓存通常被用来缩小存储级别之间的性能差距。然而，由于DRAM和Optane之间的性能差距较小，使用DRAM作为Optane的缓存将比使用NAND或HDD的好处少。MySQL InnoDB的文件缓存被称为缓冲池，我们将其配置在数据大小的3%到50%之间，使用高斯分布和混合读写访问模式来测试OLTP性能。

图12（a）和图13（a）分别显示了PE和VE中的结果。从3%的数据缓存转移到50%的数据缓存，使得PE中的Optane增加了其TPS增加了1.4倍，PE中的NAND增加了1.9倍，VE中的Optane增加了1.3倍，而VE中的NAND增加了1.5倍。因此，无论是在VE还是在PE中，缓存对于速度较慢的NAND来说都更为重要。

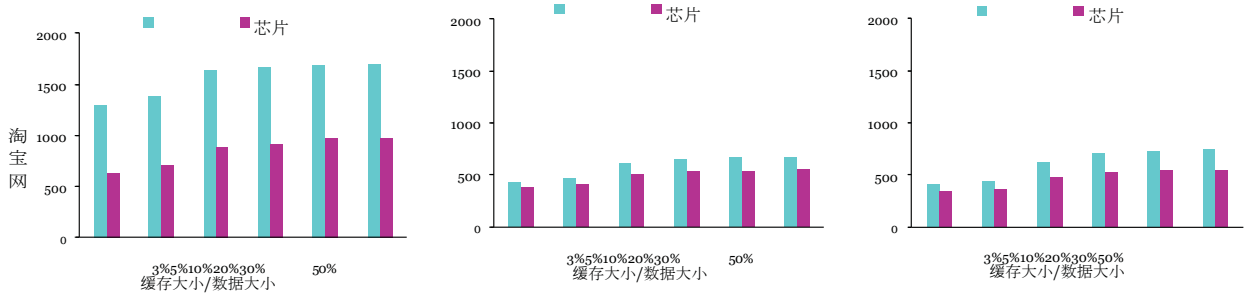


(a) 未经压缩的表格（页面大小为16KB）。

(b) 压缩表（页面从16KB压缩到8KB）。

(c) 压缩表（页面从16KB压缩到4KB）。

图12：MySQL OLTP在PE中使用不同缓存大小的性能。



(a) 未经压缩的表格（页面大小为16KB）。

(b) 压缩表（页面从16KB压缩到8KB）。

(c) 压缩表（页面从16KB压缩到4KB）。

图13：在VE中使用不同缓存大小的MySQL OLTP性能。

D. 表的压缩

我们在第四节讨论了I/O密集型应用的透明压缩。许多数据库存储引擎，包括MySQL InnoDB，都有一个压缩功能[23][24]。MySQL InnoDB支持的页级压缩被称为表压缩。当它被启用时，InnoDB将尝试把每个页面压缩成更小的页面，例如把16KB的页面压缩成4KB的页面。压缩后的页面将被存储在文件中或缓存在缓冲池中。

压缩引起的性能提升只能在使用慢速HDD时发生，而在SSD中使用表压缩的目的只是为了节省存储空间。如图12(b)(c)和图13(b)(c)所示，我们设置压缩后的页面大小为8KB和4KB，测试PE和VE中缓存大小增加时的OLTP性能。与图12（a）和图13（a）中未压缩的情况相比，使用表压缩功能略微降低了配备MySQL的NAND的性能，而配备MySQL的Optane的性能下降是显著的。

因此，在MySQL InnoDB中，配置表压缩功能是对存储空间的节省和Optane与NAND在PE中的性能的权衡。在VE中，使用压缩的成本很高，以至于Optane的性能下降到NAND的水平。

VI. 相关的工作

许多研究工作已经评估了新兴固态硬盘的性能。Xu等人[25]对NVMe驱动器进行了首次深入的性能分析，比较了HDD、SATA SSD和NVMe SSD。Son等人[15]在Linux文件系统和数据库工作负载的背景下评估了NVMe SSD的性能，对NVMe SSD和SATA SSD进行了比较。Hady等人[12]介绍了3D XPoint技术，并使用Optane评估了其性能，主要关注基于3D XPoint的设备的使用情况。Wu等人[26]评估了Optane高性能计算（HPC）应用，在Optane和HDD之间进行了比较。在本文中，我们对基于NAND闪存的SATA SSD和基于3D XPoint的Optane SSD之间的对比更感兴趣。我们认为SATA SSD和PCIe 3D XPoint SSD之间的对比是有意义的，因为SATA连接的SSD今天在许多数据中心仍然被广泛使用，而PCIe接口的NVMe协议正变得越来越流行。

随着云计算的不断发展，存储设备的速度也越来越快，人们已经开始着手优化VMM的存储栈。QEMU过去只支持一个I/O线程来满足所有虚拟机的I/O请求，因此造成了I/O可扩展性问题。后来，这个问题通过virtio-blk-dataplane功能得到了改善，它为每个设备使用了一个专用的I/O线程[27]。但随着NVMe SSD的出现

进入市场后，QEMU的I/O瓶颈再次出现，因此也做了很多优化，例如在每个设备上使用多个I/O线程[28] [29]，使用用户空间的NVMe驱动[30]，以及轮询异步I/O完成[31]。随着Optane固态硬盘进入市场，我们相信获得Optane在最先进的虚拟化环境中的表现的知识对于准备使用新设备的云数据中心是有意义的。

3D XPoint DIMM的形式，即所谓的Optane DC持久性内存（PM）[32]，据说将在不久的将来广泛使用。由于存储和内存的模糊性，PM将极大地改变存储架构和硬件及软件的编程模型，这将是一个长期的演变。在本文中，我们不讨论基于3D XPoint的PM设备，而是专注于最近发布的技术。

VII. 结论

基于3D XPoint的Optane SSD在各方面都优于基于NAND闪存的SSD，如更低的延迟、更高的并行性和更高的带宽。我们全面分析了3D XPoint SSD的性能特征。根据分析，传统的优化措施，如文件缓存、透明数据压缩，要么变得不那么有效，要么甚至降低新设备的性能。由于目前的VMM对这些高性能存储设备的支持较差，因此在虚拟化环境中情况更为复杂，需要进一步研究。我们未来的工作将根据这些特征对操作系统、VMM和I/O密集型应用进行专门的优化。

鸣谢

这项工作得到了国家自然科学基金（61602266，61872201，U1833114）、天津市科技发展计划（17JCYBJC15300，16JCY-BJC41900）、中央高校基础研究基金和SAFEA：文化教育领域海外青年人才的部分支持。

参考文献

[1] S.R. Dulloor，*持久性记忆的系统和应用*。博士论文，乔治亚理工学院，2015。
[2] "NVMe." https://www.nvmexpress.org/。[在线；访问10-July- 2018]。
[3] M.Bjørling, J. Axboe, D. Nellans, and P. Bonnet, "Linux block IO: introducing multi-queue SSD access on multi-core systems," in *Proceedings of the 6th International Systems and Storage Conference*, p. 22, 2013.
[4] P.Bonnet, "What's up with the storage hierarchy?", in *Conference on Innovative Data Systems Research*, 2017.
[5] K.Kant，"数据中心的演变。A tutorial on state of the art, issues, and challenges," *Computer Networks*, vol. 53, pp.2939-2965, 2009.
[6] "灵活的I/O测试仪。" https://github.com/axboe/fio。[在线；2018年7月10日访问]。
[7] A.D. Brunelle, "Block I/O layer tracing: blktrace," in *Gelato-Itanium Conference and Expo*, 2006.
[8] N.Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. S. Manasse, and R.Panigrahy，"SSD性能的设计权衡"，在*USENIX年度技术会议论文集*，第57-70页，2008。
[9] "Intel SSD DC S3510 Series Specifications." https://ark.intel.com/products/86197/Intel-SSD-DC-S3510-Series-480GB-2_5in-SATA-6Gbs-16nm-MLC.[Online; accessed 10-July-2018].
[10] A.Gulati, G. Shanmuganathan, I. Ahmad, C. Waldspurger, and M.Uysal, "Pesto: online storage performance management in virtualized datacenters," in *Proceedings of 2nd ACM Symposium on Cloud Computing*, p. 19, 2011.
[11] J.Basak和M. Bharde，"存储工作负载的动态配置"，在*第29届大型安装系统管理会议论文集*，第13-24页，2015年。
[12] T. Hady, A. Foong, B. Veal, and D. Williams, "Platform storage performance with 3D XPoint technology," *Proceedings of the IEEE*, vol. 105, pp. 1822-1833, 2017.
[13] "ZFS." https://en.wikipedia.org/wiki/ZFS。[在线；访问10-July-2018]。
[14] Y.Son, H. Kang, J.-Y.Ha, J. Lee, H. Han, H. Jung, and H. Y. Yeom, "An empirical evaluation of enterprise and SATA-based transactional solid-state drives," in *Proceedings of the 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp.231-240, 2016.
[15] Y.Son, H. Kang, H. Han, and H. Y. Yeom, "An empirical evaluation and analysis of the performance of NVM express solid state drive," *Cluster Computing*, vol. 19, pp. 1541-1553, 2016.
[16] T.Makatos, Y. Klonatos, M. Marazakis, M. D. Flouris, and A. Bilas, "Using transparent compression to improve SSD-based I/O caches," in *Proceedings of the 5th European Conference on Computer systems*, pp.1-14, 2010.
[17] W.Xia, H. Jiang, D. Feng, F. Douglis, P. Shilane, Y. Hua, M. Fu。Y.Zhang, and Y. Zhou, "A comprehensive study of the past, present, and future of data deduplication," *Proceedings of the IEEE*, vol.104, pp.1681-1710, 2016.
[18] Khan, R. C. Burns, J. S. Plank, W. Pierce, and C. Huang, "Rethinking erasure codes for cloud file systems: minimizing I/O for recovery and degraded reads.", in *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, p. 20, 2012.
[19] "NTFS。"https://en.wikipedia.org/wiki/NTFS。[在线；访问10-July-2018]。
[20] "MySQL。" https://www.mysql.com/。[在线；2018年7月10日访问]。
[21] "lzbench." https://github.com/inikep/lzbench。[在线；访问10-July-2018]。
[22] A.Kopytov, "Sysbench:一个系统性能基准。" https://github.com/akopytov/sysbench, 2004.
[23] S.Aghav, "用于性能优化的数据库压缩技术"，在*第二届国际计算机工程与技术会议论文集*，第6卷，第V6-714页，2010年。
[24] J.Ma, B. Yin, Z. Kong, Y. Ma, C. Chen, L. Wang, G. Wang, and X. Liu, "Leveraging page-level compression in MySQL - a practice at baidu," in *Proceedings of the 14th IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp.1085-1092, 2016。
[25] Q.Xu, H. Siyamwala, M. Ghosh, T. Suri, M. Awasthi, Z. Guz, A.Shayesteh和V. Balakrishnan, "NVMe SSD的性能分析及其对现实世界数据库的影响，"在*第八届ACM国际系统和存储会议论文集*，第6页，2015。
[26] K.Wu, F. Ober, S. Hamlin, and D. Li, "Early evaluation of Intel Optane non-volatile memory with HPC I/O workloads, " *arXiv preprint arXiv:1708.02199*, 2017.
[27] S.Hajnoczi, "Towards multi-threaded device emulation in QEMU," in *KVM论坛*，2014年。
[28] T.Y. Kim, D. H. Kang, D. Lee, and Y. I. Eom, "Improving performance by bridging the semantic gap between multi-queue ssd and I/O virtualization framework," in *Proceedings of the 31st IEEE Symposium on Mass Storage Systems and Technologies*, pp.
[29] D.Zhang, H. Wu, F. Xue, L. Chen, and H. Huang, "High performance and scalable virtual machine storage I/O stack for multicore systems," in *Proceedings of the 23rd IEEE International Conference on Parallel and Distributed Systems*, pp.292-301, 2017.
[30] F.Zheng, "QEMU中的用户空间NVMe驱动", in *KVM论坛*, 2017.
[31] S.Hajnoczi，"将轮询技术应用于QEMU"，在*KVM论坛*，2017年。
[32] "英特尔Optane技术。" https://www.intel.com/content/www/us/en/architecture-and-technology/intel-optane-technology.html。[Online; accessed 10-July-2018].