用于索引数据结构的 Optane DC 持久内存能耗评估

Manolis Katsaragakis*†、Christos Baloukas*、Lazaros Papadopoulos*、Verena Kantere•、Francky Catthoor†◦、Dimitrios Soudris**雅典国立技术大学电子与计算机工程系微处理器与数字系统实验室，希腊

†鲁汶天主教大学，阿伦贝格城堡公园 *10* 号，*3001* 海弗莱，比利时
•雅典国立技术大学电子与计算机工程系知识与数据库系统实验室，希腊
◦*IMEC*，卡佩尔德雷夫 *75* 号，*3001* 海弗莱，比利时*{
mkatsaragakis, lpapadop, cmpalouk, dsoudris}@microlab.ntua.gr
•vkante@dblab.ece.ntua.gr，†francky.catthoor@esat.kuleuven.be

arXiv:2304.00953v1 [cs.DB] 3 Apr 2023

摘要——**英特尔 Optane DC 持久内存（DCPM）是一项极具吸引力的新技术，可用于构建面向数据密集型高性能计算应用的存储系统。与动态随机存取存储器相比，它的单位字节成本更低、待机功耗更低、容量更大，且延迟相当。本文深入评估了 Optane DCPM 的能耗，并采用专为应对持久内存的挑战和限制而设计的成熟索引。我们研究了 Optane DCPM 在多种索引数据结构及 LevelDB 键值存储中，针对不同类型 YCSB 工作负载的能效。与典型的固态硬盘存储方案相比，在内存系统中集成 Optane DCPM 后，LevelDB 实验的能耗降低了 71.2 %，吞吐量提高了 37.3%**。

索引词——**Intel Optane DCPM、非易失性内存、索引、LevelDB、可扩展性、能耗、高性能计算**

I. 引言

内存系统是限制超级计算机可扩展性并导致其能耗增加的主要因素之一 [1]。为支持更复杂的模拟、分析和高效的内存中处理而集成更多动态随机存取存储器模块，会对超级计算中心的可持续性和维护成本产生负面影响。尤其是，尽管传统动态随机存取存储器技术的访问延迟较低，但其不断增加的漏电功耗和刷新功耗需求限制了动态随机存取存储器的可扩展性，也给实现百亿亿次级性能带来了重大挑战。

为克服动态随机存取存储器的局限性，人们引入了非易失性存储器技术，例如 3D-XPoint。它属于相变存储器（PCM） [2]、自旋转移矩随机存取存储器（STT-RAM） [3] 和电阻式随机存取存储器（ReRAM） [4]等技术。当前最先进的商用平台将英特尔 Optane DC 持久内存（DCPM）模块与动态随机存取存储器集成在一起，从而构成异构内存系统 [5]。例如，即将推出的 Aurora 百亿亿次级超级计算机采用了 DAOS 存储架构，该架构集成了复杂的内存与存储层次结构，其中包括英特尔 Optane DCPM 模块 [6]。

本研究部分经费由欧盟“地平线2020”计划通过资助协议 No101015922AI@EDGE（https://aiatedge.eu/）提供。

这些新兴存储器技术的密度高于动态随机存取存储器，因此能够以更少的节点实现更大的总内存容量，并对能耗、韧性和可持续性产生积极影响。此外，非易失性存储器技术的数据持久化特征可用于为应用程序提供容错支持。另一方面，与动态随机存取存储器相比，Optane DCPM 通常具有更高的访问延迟和更低的带宽 [5], [7], [8]。不过，近期研究表明，Optane DCPM 的性能在很大程度上取决于工作负载访问模式和规模 [8]。

近期多项研究考察了 Optane DCPM 技术在支持更复杂、高效的高性能计算模拟方面的作用 [7], [1], [9], [10], [11]。这些研究表明，将 Optane DCPM 用作易失性存储器，并将动态随机存取存储器用作缓存，可以实现接近动态随机存取存储器的性能。然而，若不将动态随机存取存储器用作缓存而直接以 Optane DCPM 替代动态随机存取存储器，性能会显著下降，原因包括写入限流和并发竞争等影响 [12]。

或者，Optane DCPM 可用作持久化存储介质，以支持存储和处理模拟或仪器收集的海量科学数据 [13], [14]。近期研究表明，为了利用持久内存所支持的存储系统为应用程序带来的可扩展性优势，需要调整存储系统的索引数据结构，以应对非易失性存储器技术带来的挑战和限制 [15], [16], [17]。例如，使用非易失性存储器进行持久化存储时，一个根本性的挑战是数据一致性。由于现代 CPU 是针对易失性 DRAM 架构设计的，为了追求更高的性能，它们通常会缓存并重新排序内存写入。因此，现有的索引结构是专为仅使用 DRAM 的系统设计的，无法直接部署在异构层次结构中，因为它们没有考虑非易失性存储器的持久特性，也没有考虑其他挑战，例如读写延迟的不对称性以及非易失性存储器有限的写入耐久性。

在这一方向上，文献中已有一些研究

提出了专为非易失性存储器设计的索引 [15], [16], [17], [18], [19]。一些近期研究对用于持久内存的 B+ 树索引数据结构进行了基于性能的评估 [20], [21], ，而另一些研究则将基于动态随机存取存储器的 B+树、字典树、基数树和哈希表索引转换为面向非易失性内存的结构，并评估其对吞吐量和缓存利用率等指标的影响 [22]。

本研究面向考虑将 Optane DCPM 用作持久化存储的高性能计算开发者，这些开发者通过索引数据结构访问大量数据。现有文献中的研究要么侧重于评估 Optane DCPM 作为易失性主存的表现，要么使用 Optane DCPM 作为存储设备，对其持久性特征进行性能表征。尽管这些研究都很有参考价值，但在整体分析中仍遗漏了重要的能耗因素。因此，为弥补这一不足，本研究着重评估将 Optane DCPM 配置为持久化存储时的能耗。本研究的两项主要贡献如下:

- 全面评估 Optane DCPM 作为存储介质时的能耗，采用具有代表性且广为认可的 B+ 树索引数据结构，并通过多种类型的 YCSB 工作负载进行测试。
- 对部署在 Optane DCPM 上的 LevelDB 键值存储进行性能与能耗的综合评估，并与部署在固态硬盘上的对应方案比较能耗和吞吐量。

据我们所知，这是首篇完全聚焦于 Optane DCPM 能效方面的研究。尽管已有多项研究探讨这种新型内存架构的性能表现（详见后续章节），但迄今为止，对 Optane DCPM 能耗特性的评估仍十分有限。

本文其余部分组织如下：第二节讨论相关工作，重点介绍本研究与现有工作的主要区别。第三节概述 Optane DCPM 技术。第四节介绍评估方法。第五节给出基于 B 树索引数据结构和 LevelDB 键值存储的一组可扩展性基准结果及能耗分析。此外，本节还总结了主要观察结果和开放研究方向。最后，第六节总结全文。

II. 相关工作

与 Optane DCPM 相关的现有研究大致可分为三类:

- 通过微基准测试和/或应用程序对 Optane DCPM 的性能进行实证评估，并探讨其面临的挑战与局限 [5], [7], [1], [9], [12], [10], [22], [20], [21]
- 针对作为持久化存储的 Optane DCPM 优化的新型应用程序和数据结构（主要是新型树型数据结构） [23], [24]
- 适用于 Optane DCPM 的新型工作负载调度器、数据放置算法及其他运行时 [25], [8], [26]

本研究主要属于第一类，旨在使用索引数据结构全面评估 Optane DCPM。不过，与现有研究不同，本研究侧重于能耗方面。

近期，研究人员使用真实世界的科学应用对 Optane DCPM 进行了广泛评估。这些研究采用的典型硬件配置是将 Optane DCPM 用作主存，而将动态随机存取存储器用作缓存。在这种情况下，Optane DCPM 的持久化特性无法发挥作用。近期研究得出结论：在这种硬件配置下，Optane DCPM 与纯 DRAM 系统之间的性能差距相对较小 [12], [9]。完全以 Optane DCPM 替代动态随机存取存储器（即完全禁用动态随机存取存储器）的研究指出，对于真实世界的多线程应用程序而言，Optane DCPM 的带宽饱和会导致显著的性能开销 [5], [1]。

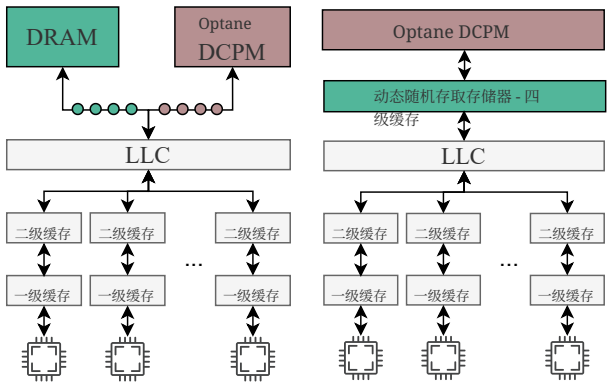
与上述研究不同，将 Optane DCPM 用作持久性内存需要修改应用程序源代码。此外，用于将数据存放到 Optane DCPM 上的 API 仅适用于动态分配的数据结构（例如通过 *libvmmalloc*） [7]。因此，现有文献仅使用真实世界的科学应用程序对这种配置进行了部分评估 [10], [12]。大多数研究通过微基准测试（如 Stream Triad）或小型应用程序来评估这种配置下的 Optane DCPM [7]。文献中有一项更全面的评估，同时考察了性能和能耗，但它专门针对图应用程序 [12]。另一方面，也有多项研究将 B+ 树索引数据结构作为持久性内存部署在 Optane DCPM 上进行评估。这些研究通常依据吞吐量和延迟等应用级性能指标来比较索引 [22], [20], [21]。

总而言之，与现有研究不同，本研究使用 B+ tree 索引，从能耗角度评估 Optane DCPM 作为持久性内存的表现，同时还对 LevelDB 键值存储进行了性能与能耗的综合分析。

III. Optane DCPM 概述

Intel Optane DCPM 是首款可商用的非易失性双列直插式内存模块。它基于 3D-XPoint 技术，同时具备字节可寻址性和持久性。从中央处理器的角度来看，访问粒度为 64 字节。然而，物理介质的访问粒度为 256 字节。因此，小于 256 字节的写入访问属于读-修改-写操作，进而导致写入放大。文献 [10] 已对写入放大的影响进行了广泛研究。

Optane DCPM 的容量远高于动态随机存取存储器（ DRAM），功耗更低；但通常代价是延迟增加、带宽降低。图 1 描绘了 Optane DCPM 在不同替代运行模式下的异构内存系统:



（a）AppDirect 模式　　（b）内存模式

图1：Intel Optane DCPM 工作模式概览

Optane DCPM。Optane DCPM 可配置为支持 *AppDirect* （图 1a）或 内存模式（图 1b）。在 AppDirect 模式下，Optane DCPM 作为持久化存储运行。访问存储空间的方式类似于内存映射文件操作。在内存模式下，Optane DCPM 作为主内存空间运行，不具备持久化属性。在此模式下，动态随机存取存储器（DRAM）用作末级缓存之上的额外缓存层。

在*AppDirect* 模式下，开发人员可借助持久内存开发套件（PMDK）进行持久内存分配 [27]。该套件提供了 *libvmmalloc* 和*pmemobj* 等多个库，用于动态分配通常由 *malloc*、*free* 和*memalign*处理的对象。例如，*libv mmalloc* 可在运行时将所有动态分配透明地映射到持久性内存中。因此，动态分配的数据结构会存放在内存映射文件中，而该文件存储于持久性内存中。不过，在内存模式下，默认情况下所有数据对象都会分配到 Optane DCPM 中，这一过程对开发人员透明。本研究将 Optane DCPM 配置为*AppDirect* 模式，旨在将其用作数据密集型应用的持久化存储，而非易失性存储器。

IV. 评估方法

为了选择针对持久性内存设计的代表性B+ 树索引实现，我们首先根据相关文献确定了它们的主要设计目标。这些目标概述如下：

- **最小化写操作**。Optane DCPM 中的写操作在延迟方面代价高昂。因此，为持久性内存开发的索引采用了一系列方法来最小化写操作，例如在执行原子操作时减少日志记录，或避免使用通常为维护持久性内存中数据的一致性而需要进行的 CPU 缓存刷新。

- **高效的锁机制**。为了实现高并发，读操作应以一致且高效的方式在多个线程之间执行，从而尽可能充分地利用可用带宽，

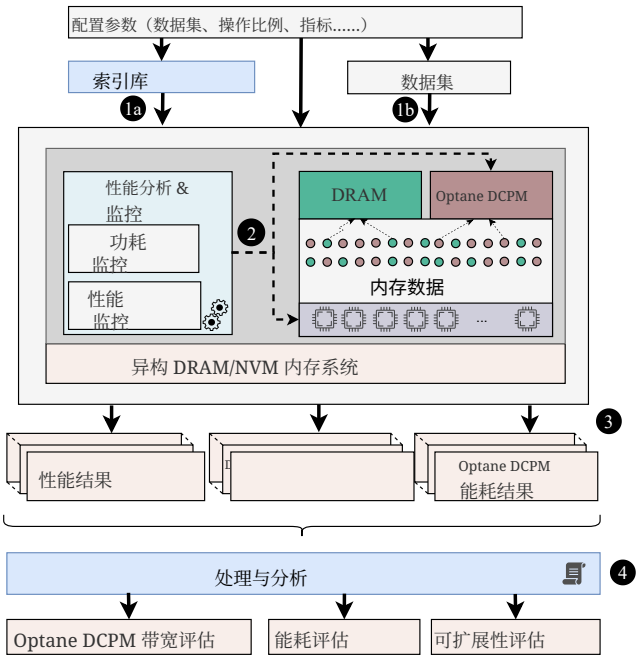


图 2：评估方法概览

另一方面，写操作应尽可能使用细粒度的锁，将临界区限制在必要的代码范围内。

- **高效的索引遍历**。排序、树的类型、键分裂等多项决策对索引遍历起着重要作用。一些索引的实现旨在提高数据局部性，而另一些则将树的一部分存储在动态随机存取存储器中，以实现快速遍历。

这些目标在不同的索引实现中以不同方式体现，因为它们都试图以高效的方式提供持久性和并发。已有许多不同的实现方案，包括树、哈希表、字典树、基数树等。 [22]。在本文中，我们重点关注树实现，因为它们在实践中最为常用。

我们设计了一套系统化的方法，以便有效运行文献中广泛使用的四种索引，如图 2 所示。该方法包含四个不同的步骤。方法的输入是一组索引　和一个数据集　，它们由一组参数配置，例如工作负载类型（操作比例、偏斜度等）、用于评估的索引以及关注的指标（如吞吐量、能耗、可扩展性）。索引库基于 [22] 提供的源代码，+ 其中包含下文列出的 B 树实现。这些索引的选择依据是：它们能够代表实现上述设计目标的不同方式。

- **wBTree** 是一种主存 B+ 树，旨在减少大量非易失性内存写入和中央处理器缓存刷新操作所带来的开销。为了最大限度地减少写入

操作，wBTree 使用支持原子写入的 B+树，可通过节点中的原子写入或仅重做日志实现节点一致性，从而减少在 Optane [28] 上所需的写操作。此外，执行过程中键始终按顺序排列，节点采用小型间接寻址槽数组和/或位图，以快速检查特定键是否存在。使用间接寻址槽数组还带来了额外好处：大多数插入和删除操作无需移动索引条目，从而进一步减少写操作的数量。

- **Masstree：**是一种缓存高效、支持高并发的树形拼接结构，由B+ 树节点构成，即使面对较长的公共键前缀，也能提供出色的性能。我们使用了 [22]针对非易失性内存改造的Masstree版本。Masstree结合了字典树和B+树实现，以提升长键的处理性能。具体而言，较长的字符串键会拆分到多个节点中，因此搜索速度更快，因为键的公共部分可以更快地进行比较。Masstree采用版本控制系统来实现并发，从而提升锁定性能。此外，为减少写操作，Masstree会将新键以未排序的顺序追加到节点中的新键值对，从而插入叶节点。内部节点则保持排序，以便更快地遍历。

- **Fast&Fair：**是一种持久性内存B+树，可提供无锁读取 [22]。读取过程能够检测并容忍重复元素等不一致情况。通过使读操作能够容忍瞬态不一致性，Fast& Fair避免了开销高昂的写时复制、日志记录，甚至无需使用读闩锁，因此读事务可以采用非阻塞方式执行 [29]。Fast&Fair的这些特性减少了写操作，同时提升了锁定性能。

- **FPTree：** 是一种 B+Tree 替代方案，通过仅要求叶节点持久化来实现选择性持久化。发生断电故障时，内部节点可以重建。因此，FPTree 将所有内部节点放在动态随机存取存储器中，以便快速遍历并获得更高的性能，同时将叶节点存放在持久性内存中。此外，叶节点采用指纹化，即将节点所含每个键的单字节哈希值存放在节点首个缓存行大小的区域中 [16]。指纹化通过加快查找键是否包含在特定节点中的过程来提升查找性能，无需遍历节点本身。在并发方面，FPTree 对内部节点和叶节点分别使用硬件事务内存（HTM） [30] 和细粒度锁。

该性能分析与监控组件（ ）负责功[illegible]与性能监控，基于英特尔处理器计数器监控器（PCM） [31], 这是一款通过硬件计数器对双列直插式内存模块进行能耗/功耗采样的工具 [32]。采样率

表 I：实验设置概览

实验设置	
中央处理器、动态随机存取存储器	至强 Gold 5218R 中央处理器，2×20 个核心 @2.10GHz，配备 4x32GB DDR4 双列直插式内存模块
Optane DCPM 配置	2x256GB 双列直插式内存模块，AppDirect 模式，ext4-DAX
SSD	英特尔 SSD S4500 系列，480GB
索引	wBTree、masstree、Fast&Fair、FPTree
键值存储	LevelDB
YCSB 工作负载	均衡型（50% 读取，50% 写入）、写密集型（99% 写入），读密集型（100% 读取）、扫描密集型（99% 扫描）
工作负载键	8 字节整数
操作系统	Ubuntu 20.04.2 LTS，内核 5.4.0-121-generic

性能分析的默认采样间隔为 0.01 秒。具体而言，我们使用*PCM-Power* 组件监控动态随机存取存储器和 Optane DCPM DIMM 的能耗。对于每个独立的双列直插式内存模块，*PCM-Power* 会根据我们设定的采样率报告该时间段内的内存模块能耗。Optane DCPM 上的内存访问由 *ipmctl* 实用程序测量。该工具是配置和管理 Optane DCPM 模块的标准工具，分别监控每个双列直插式内存模块对应的读取和写入访问 [33]。最后，为了计算固态硬盘的能耗，我们使用 Intel VTune Amplifier 报告的输入/输出性能分析信息（包括页面读取、写入和刷新操作），以及制造商数据手册中的功耗值。

这些测量结果会被分组（ ），[illegible]并按顺序处理（ ）[illegible]以得出每项评估实验的吞吐量、能耗和可扩展性结果。上述结果由一组可定制的 Python 和 Shell 脚本自动生成。该工具的源代码、使用说明以及索引库均公开可用 [1]。

V. 评估

A. 实验设置

通过评估过程，我们旨在实现以下目标:

- 评估 Optane DCPM 在各类工作负载下用于索引数据结构时的能耗表现。

- 研究 OptaneDPCM 用于键值存储（LevelDB）时的性能和能效，并以部署在固态硬盘上的相应方案作为基准。

表 I 展示了实验设置概况。实验在单节点上进行，配备 2 颗 20 核、主频为 2.10 GHz 的 Intel Xeon Gold 5218R 中央处理器、4 根 32 GB DDR4 双列直插式内存模块和 2 根 256 GB Optane DC 非易失性双列直插式内存模块。Intel Optane DC 配置为 *App Direct* 模式，并使用 ext4-DAX 文件系统。我们使用了 PMDK 1.11 版 [27] 和 gcc-9.4。监控和性能分析工具已在第 IV 节中介绍。

使用 YCSB 在不同类型的工作负载下触发索引：均衡型、写密集型（对应插入操作）、读密集型（即查找操作）以及

[1]https://github.com/mkatsa/PENVMTool

扫描密集型（即范围查找操作）[34]。每种工作负载包含 6400 万次操作，使用 8 字节的随机整数键。

B. 基准性能结果

首先，我们展示一组基准性能结果，以便全面了解后续的能耗评估。

图 3 展示了 Optane DCPM 上多种 YCSB 工作负载的吞吐量，并说明了各索引在不同工作负载下的可扩展性。可以看出，在读密集型和扫描密集型工作负载下，*fptree* 可扩展至 32 个线程，吞吐量分别超过 10Mops/sec 和 5 Mops/sec。对于均衡型和写密集型工作负载，*masstree*、*fptree* 和*fastfair* 在 16 个线程时的最大吞吐量分别可达 3.3Mops/sec 和 2.3Mops/sec。*wbtree* 无法扩展，因为本研究采用的实现为单线程实现；它作为其余 B+ tree 实现的顺序基准。

作为比较，我们评估了索引在纯DRAM内存系统上的可扩展性。这些基准结果如图 4 所示。所有索引，尤其是 *masstree*，在均衡型工作负载下扩展性良好，最高可支持 64 个线程。此外，我们观察到，在纯 DRAM 系统上，*masstree* 索引的吞吐量高于*fptree* ；而在 Optane DC 执行环境中则相反，*fptree* 的性能优于*masstree*。这是因为*fptree* 将所有内部节点放在 DRAM 上，仅将叶节点放在 Optane 上，因此在更多线程下性能更好，尤其是在只读型工作负载下。对于其余工作负载类型，我们也观察到了类似结果。

为研究 Optane DCPM 硬件配置对可达到的最大吞吐量的影响，我们在均衡型工作负载下，评估了索引在仅配备一个 Optane DCPM DIMM 的内存系统上的可扩展性。对比图 3a（两个 DIMM）与图 5（单个 DIMM）的结果可以发现，在同一处理器插槽上使用更多 Optane DCPM DIMM 可提高并行性，从而改善可扩展性并提升吞吐量。事实上，在单个 DIMM 的情况下，最大吞吐量在 4 个线程时达到，且仅略高于 1Mops/sec。

上述观察结果与现有文献中的性能评估结果一致。特别是，近期研究借助定制微基准测试 [20]，对 Optane DCPM 上 B+ 树索引的可扩展性限制进行了研究。我们基于 YCSB 得出的结果进一步证实了观察到的可扩展性限制。

C. 能耗结果

图 6 展示了不同类型工作负载下各索引在 Optane DCPM DIMM 中消耗的能量，该数据由 DIMM 配备的传感器测得。总体而言，能耗逐渐降低，直至线程数达到 8 或 16，这表明所达到的吞吐量与消耗的能量直接相关。*wbtree*索引的能耗则基本保持不变。由于 wbtree 无法扩展（图 3），内存资源会被占用更长时间，

表 II：16 个线程下，各工作负载类型对应的 Optane DCPM DIMM 能耗（KJ）。

	均衡型	写密集型	读密集型	Scan
wbtree	3.539	4.069	2.966	4.372
masstree	0.615	0.701	0.524	0.645
fptree	0.554	0.627	0.465	0.562
fastfair	0.787	0.896	0.620	0.738

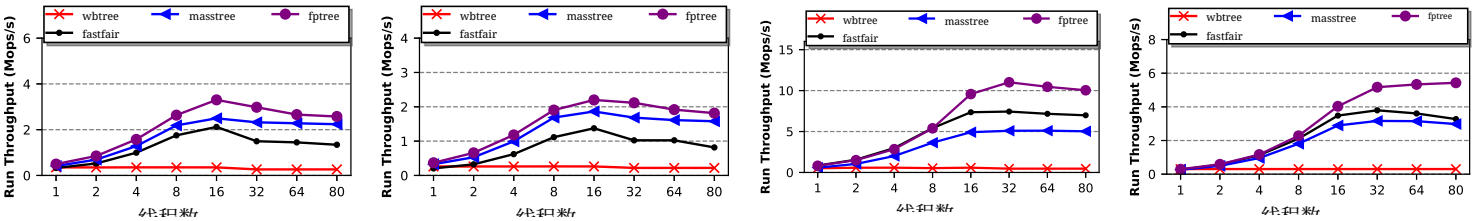
因此导致能耗增加。

表 II 展示了 16 个线程下的能耗值。我们注意到，在相同工作负载下，不同索引的能耗差异并不显著。然而，与其他工作负载类型相比，Optane DCPM 在读密集型工作负载下能耗更低。例如，在读密集型工作负载的作用下，*fptree*索引的能耗比相应的均衡型工作负载低 16%，这表明读密集型工作负载下吞吐量的提升会影响能耗（图 3）。

作为 Optane DCPM 能耗的基准，图 7 展示了仅使用动态随机存取存储器系统（即停用 Optane DCPM）时均衡型工作负载的能耗。与图 6 类似，结果仅对应于 DRAM DIMM 上消耗的能量。随着线程数达到 4 或 8，能耗显著降低，这与图 4 所示的吞吐量提升相符。此外，通过比较图 7 和图 6a 的能耗值，我们发现 Optane DCPM 中消耗的能量约为 DRAM 的一半。例如，在 Optane DCPM 上部署 *fptree* 并使用 16 个线程时，能耗比相应的 DRAM 实验低 56.3%。

图 8 展示了总能耗，即中央处理器和内存系统的能耗。吞吐量仍会影响能耗：线程数为 8–16 时能耗最低，此时达到最大吞吐量。然而，在此之后，对于大多数索引而言，线程数超过 16 后吞吐量下降，导致能耗略有增加，如图 3 所示。

为了进一步研究随着线程数量增加，Optane DCPM 的能耗变化，我们针对不断增加的线程数，监测每个索引在持久性内存中的读写访问次数。性能分析结果如图 9 所示。值得注意的是，当线程数超过 16 个时，读写访问次数显著增加，这是由索引的内部机制和并发开销所致，但并未提升性能，如图 3a 所示。具体而言，当带宽受到限制时（例如仅使用 1 个或 2 个 Optane DCPM 双列直插式内存模块），每个索引都会花费大量 CPU 时间，尝试获取对受保护区域的独占访问、访问用于锁定和同步的索引内部变量，从而增加总访问次数。因此，线程数超过这一临界点会对吞吐量和能耗产生负面影响。



（a）均衡型工作负载　（b）写密集型工作负载　（c）读密集型工作负载　（d）扫描密集型工作负载

图 3：针对不同类型工作负载在 Intel Optane DCPM 上进行的可扩展性评估。

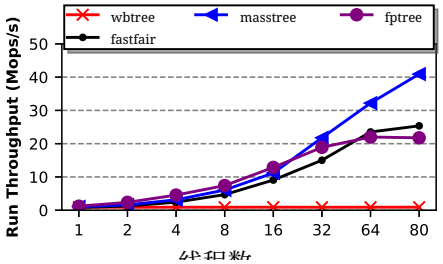


图 4：均衡型工作负载在纯DRAM上的可扩展性评估。

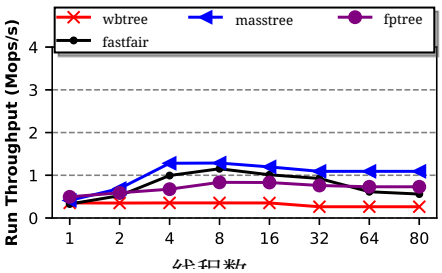


图 5：使用单个 Optane DCPM 双列直插式内存模块时，均衡型工作负载的吞吐量。

此外，内存访问次数的增加预计会影响持久性内存硬件 [35] 的耐久性。有趣的是，现有文献尚未评估 Optane DCPM 的写入硬件耐久性。

最后，为了研究可用持久内存带宽对能耗的影响，图 10 展示了集成单个 Optane DCPM 双列直插式内存模块时，均衡型工作负载的能耗。一个有趣的观察是，尽管集成 2 个双列直插式内存模块时，吞吐量几乎提升了 2 倍（如图 3a 和图 5 所示），能耗却降低了 5 倍（例如，对于 *masstree*，对比图 6a 和图 10 的结果即可看出）。因此，可以合理预期，集成更多双列直插式内存模块以提高 Optane DCPM 带宽，将进一步降低能耗并提升吞吐量。

D. LevelDB 键值存储的性能与能耗探索

在本节中，我们使用 LevelDB 对 Optane DCPM 进行性能与能耗综合分析

键值存储[2]。

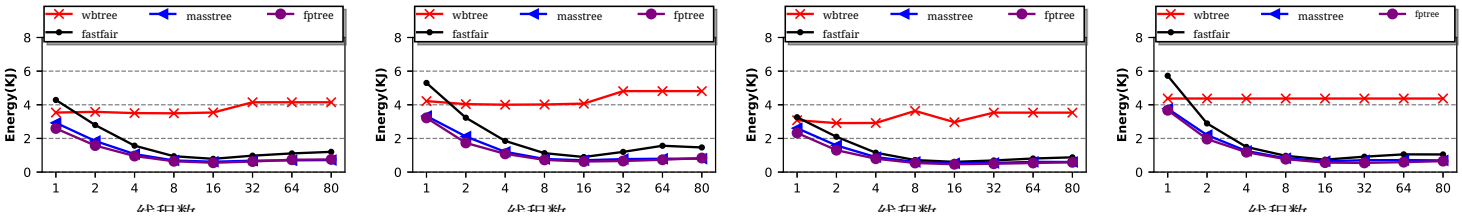
LevelDB 是 Google 开发的一种快速键值存储库，其中的键和值可以是任意字节数组，因此可以按键对任何类型的数据进行排序。LevelDB 不是 SQL 数据库，也不提供 SQL 查询接口，因此它与本文介绍的其他索引更为相似。唯一的区别是，LevelDB 将数据存储在磁盘上，同时在内存中维护一个可配置的缓冲区，以提升性能。它原生支持在 HDD/固态硬盘上进行数据持久化，并在动态随机存取存储器中进行数据缓存，因此适合用于展示将整个数据库存储在 Optane DCPM 中而非固态硬盘上的优势。LevelDB 使用日志结构合并树（也称为 LSM 树）作为数据结构来存储数据。LSM 树由两部分组成：内存中的一棵树用作缓冲区，磁盘上的另一棵树则用于存储数据。LSM 树的优势和独特之处在于，可以针对其所在的平台分别配置这两部分。

将 YCSB 集成到 LevelDB 后，我们根据图 11 所示的两种内存层次结构配置，在不同工作负载下评估其吞吐量和能耗。第一项实验中，LevelDB 存储在固态硬盘上；第二项实验中，它分配在 Optane DCPM 上。性能分析和监控所用的工具已在第 IV 节中介绍。为简洁起见，我们仅展示均衡型和读密集型工作负载的结果，但观察到的现象同样适用于写密集型和扫描型工作负载。

图 12a 和图 12b 分别展示了均衡型和读密集型工作负载下的吞吐量结果。我们注意到：

- Optane DCPM 在均衡型工作负载下的最大吞吐量出现在 16 个线程时，在读密集型工作负载下则出现在 32 个线程时。这一结果与基于 Optane DCPM 上索引数据结构实验（图 3a 和图 3c）得出的观察一致。对于 LevelDB，线程数超过这些节点后，在 Optane DCPM 上的性能会下降，这与大多数使用 B+树索引的相关实验中观察到的情况一致（图 3）。
- 当线程数超过最大吞吐量对应的线程数时，Optane DCPM 上的内存访问次数会随之增加。事实上，对于

[2]LevelDB：https://github.com/google/leveldb



(a) 均衡型工作负载　(b) 写密集型工作负载　(c) 读密集型工作负载　(d) 扫描密集型工作负载

图 6：Intel Optane DCPM 在不同类型工作负载下的能耗评估。

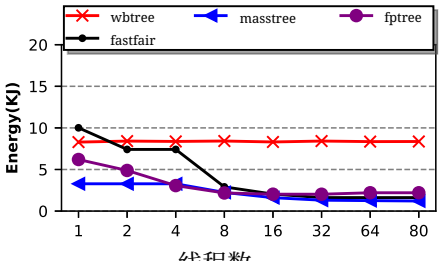


图 7：纯DRAM在均衡型工作负载下的能耗。

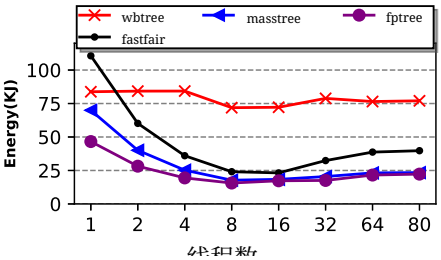


图 8：中央处理器与内存系统的能耗。

均衡型工作负载，与 16 个线程时的访问次数相比，32 个线程时的访问次数增加了 27.4%，64 个线程时则增加了 39.1%。类似地，对于读密集型工作负载，与 32 个线程时相比，64 个线程时的访问次数增加了 13.6%。这些结果与图 9 中索引数据结构实验所展示的结果相对应。内存访问次数的增加归因于同步机制带来的开销，预计会对 Optane DCPM 硬件的耐久性产生负面影响。

- 最后，图 12a 和图 12b 展示了将 LevelDB 存储在 Optane DCPM 而非固态硬盘中所能获得的性能提升。对于均衡型工作负载（16 个线程），吞吐量提升达到 37.3%；对于读密集型工作负载（32 个线程），则提升 34.6%。

均衡型和读密集型工作负载的能耗结果分别如图 13a 和图 13b 所示。文中给出了以下方面的详细测量结果：(i) CPU、DRAM DIMM 和固态硬盘；(ii) 仅 Optane DIMM；(iii) CPU、DRAM 和 Optane DIMM。我们观察到以下现象：

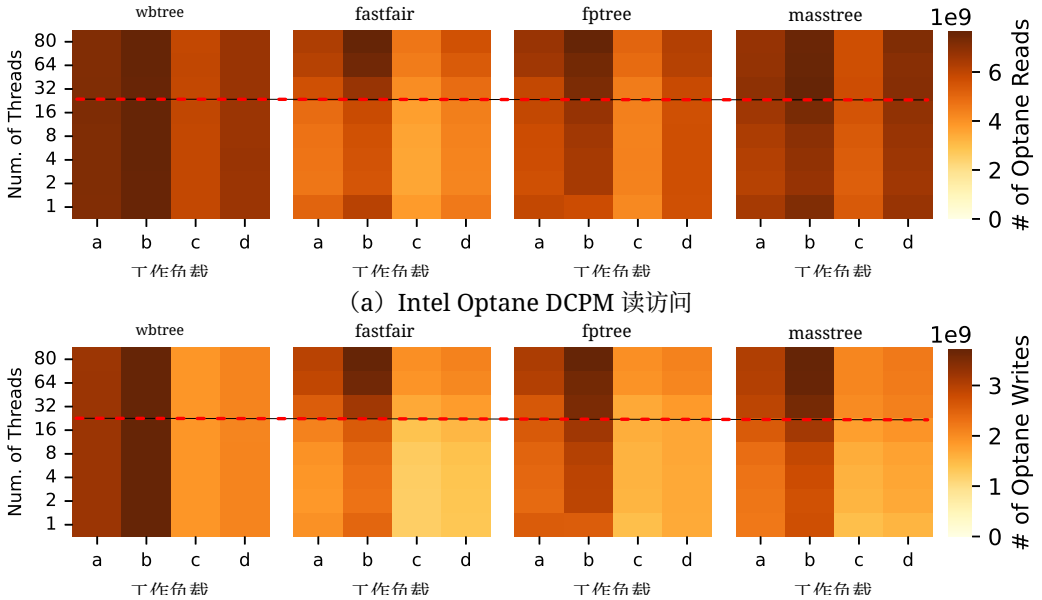
- 随着线程数增加，Optane DCPM 实验（即 CPU、DRAM 和 Optane DCPM）中均衡型工作负载的总能耗略有下降，在 8 个线程时达到 2.6KJ 的最低值，比单线程时的能耗低 16%。Optane DCPM DIMM 的能耗也与吞吐量直接相关（图 12a）。此后，随着吞吐量下降，能耗随之增加。与 Optane DCPM 实验相比，当 LevelDB 存储在固态硬盘中时，均衡型工作负载下的能耗与线程数相对独立；在 16 个线程时略有下降，与单线程实验相比降低了 2.1%。

- 对于读密集型工作负载，Optane DCPM 实验（即中央处理器、动态随机存取存储器和 Optane DCPM）的总能耗在 32 个线程时降至 0.7 千焦，此时吞吐量达到最大值。与单线程实验相比，能耗降低了 31%。仅 Optane DIMM 的能耗也在 32 个线程时降至最低。此外，将 LevelDB 存储在固态硬盘中时，能耗随线程数增加而下降，直到 32 个线程；此后则有所上升。另一个值得注意的现象是，不同工作负载类型的实验中，Optane DCPM 的能耗大致相同（均衡型和读密集型工作负载均约为 0.5 千焦）。如图 13b 所示，对于读密集型工作负载，中央处理器和 DRAM DIMM 的能耗为 0.4 至 0.6 千焦，具体取决于线程数。然而，对于均衡型工作负载，中央处理器和动态随机存取存储器消耗的总能耗占比要大得多（约为 2.5 千焦，几乎是仅 Optane DCPM DIMM 能耗的 6 倍，如图 13a 所示）。

- 将 LevelDB 存储在 Optane DCPM 上，而非固态硬盘上，可显著降低能耗。例如，对于读密集型工作负载，能耗最多可降低 71.2%。这表明 Optane DCPM 作为存储介质具有较高的能效。

E. 讨论

在本小节中，我们根据对实验结果的分析，着重介绍主要发现，并指出



（a）Intel Optane DCPM 读访问

（b）Intel Optane DCPM 写入访问

图 9：Intel Optane DCPM 的内存读写访问次数。工作负载：a：均衡型，b：写密集型，c：读密集型，d：扫描密集型。

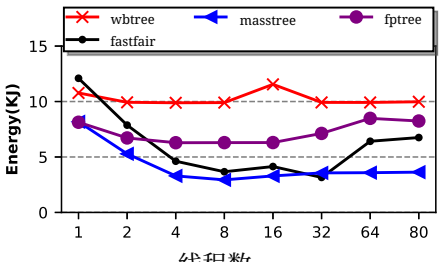
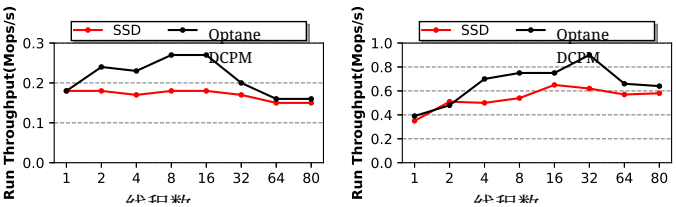
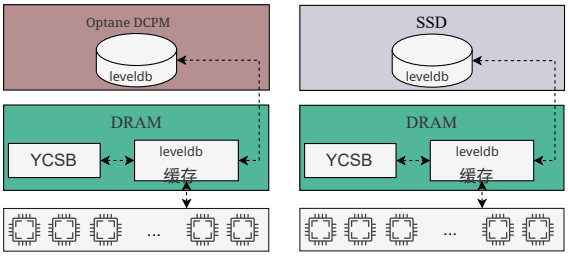


图 10：均衡型工作负载下，单个 Optane DCPM 双列直插式内存模块的能耗。



（a）均衡型工作负载　　（b）读密集型工作负载

图 12：均衡型工作负载和读密集型工作负载下，不同内存配置的 LevelDB 吞吐量。



(a) 动态随机存取存储器/固态硬盘配置。　(b) 动态随机存取存储器/Optane DCPM 配置。

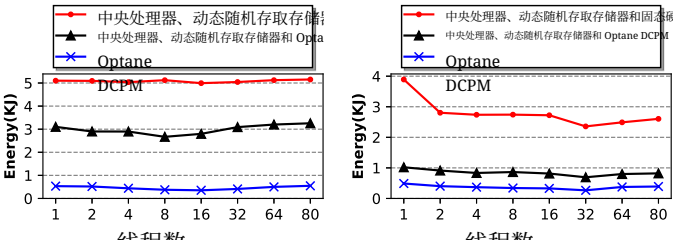
图 11：LevelDB 实验的系统配置：动态随机存取存储器/固态硬盘和动态随机存取存储器/OptaneDCPM。

一些重要的开放研究方向。

Optane DCPM 是适用于数据密集型高性能计算应用的一种持久性内存架构；与典型的固态硬盘存储相比，它在吞吐量和能耗方面均具有优势。不过，仍需考虑 Optane DCPM 的带宽限制。评估结果表明，**需要对索引数据结构在异构内存系统中的部署进行微调**，才能在能耗与性能的综合表现方面取得显著提升。

硬件配置对于部署在 Optane DCPM 上的应用程序的可扩展性和能耗至关重要。根据图 5 和图 3a 的实验，我们可以认为，将更多双列直插式内存模块集成到内存系统中，可通过提升并行性来提高吞吐量。可以合理预期，吞吐量的提升将降低能耗。然而，探究是否存在一个饱和点将很有意义；超过该点后，集成更多双列直插式内存模块将无法进一步提升能效。

写入索引操作的能耗较高与只读操作（例如查找操作）相比，在 Optane DCPM 上尤其如此。不过，增加的能耗并非由 Optane DIMM 消耗，



(a) 均衡型工作负载　　(b) 读密集型工作负载

图 13：均衡型和读密集型工作负载下，不同内存配置的 LevelDB 能耗。

正如图 13（以及图 6）中的 LevelDB 实验所示。相反，这部分能耗由 CPU 消耗，这是由于同步和一致性开销所致；为在持久化设备中维持数据一致性，需要执行缓存刷新及类似指令。在 B+ tree 和 LevelDB 实验中，均可观察到这种额外能耗。面向非易失性内存的数据一致性硬件支持，预计将显著提升整个系统的能效。

尽管实验结果分析基于使用 8字节整数键进行的实验，**但使用其他类型的键也能得出类似的观察结果**。例如，在 Optane DCPM 上使用 24 字节字符串键并采用相同的实验设置时，我们发现 Optane DCPM DIMM 的能耗与相应的整数键实验呈现出相似的模式。具体而言，在均衡型和读密集型工作负载下，能耗分别随线程数增加而降低，直至线程数达到 8 和 16；此后则趋于平稳。

在第 V 节的性能分析中，我们使用 PMDK [27] 的 *libvmmalloc* 库评估了动态分配的可扩展性。我们发现 ***libvmmalloc* 无法随线程数增加而扩展**。这与现有文献中的观察结果一致 [20]。*libvmmalloc* 的这一局限性会影响 Optane DCPM 上索引数据结构的可扩展性和能耗，也会影响那些大量使用堆内存、并通过 PMDK 库部署到 Optane DCPM 上的动态应用程序的性能。我们预计，针对持久内存的定制化、可扩展动态内存分配器将有助于提升此类应用程序的性能并降低能耗。

VI. 结论

本研究为 Optane DCPM 作为持久化存储设备提供了新的见解。基于对各种类型工作负载触发的索引数据结构进行的全面评估，我们研究了 Optane DCPM 的能耗特性。我们表明，与典型的固态硬盘相比，在内存系统中有效使用集成的 Optane DCPM，可显著提升性能并降低能耗。然而，考虑到 Optane DCPM 的带宽限制，必须进行适当的配置，才能充分发挥其优势。本研究的成果可供面向高性能计算（HPC）应用程序、并以 Optane DCPM 作为持久化存储介质的开发人员使用。

参考文献

[1] O. Patil、L. Ionkov、J. Lee、F. Mueller 和 M. Lang，“使用英特尔 Optane DC 持久内存模块对高性能计算应用的 DRAM-NVM混合内存架构进行性能表征”，载于《国际内存系统研讨会论文集》，2019年，第288–303页。[2] H.-S. P. Wong、S. Raoux、S. Kim、J. Liang、J. P. Reifenberg、B. Ra- jendran、M. Asheghi 和 K. E. Goodson，“相变存储器”，《IEEE论文集》，第98卷，第12期，第2201–2227页，2020年。[3] E. K¨ult¨ursay、M. Kandemir、A. Sivasubramaniam 和 O. Mutlu，“评估STT-RAM作为一种高能效主存替代方案”，载于《2013年 IEEE 系统与软件性能分析国际研讨会（ISPASS）》。IEEE，2013年，第256–267页。[4] S.-S. Sheu、K.-H. Cheng、M.-F. Chang、P.-C. Chiang、W.-P. Lin、H.-Y. Lee、P.-S. Chen、Y.-S. Chen、T.-Y. Wu、F. T. Chen 等，“面向嵌入式应用的快速写入电阻式随机存取存储器（RRAM）”，《IEEE计算机设计与测试》，第28卷，第1期，第64–71页，2020年。[5] J. Izraelevitz、J. Yang、L. Zhang、J. Kim、X. Liu、A. Memaripour、Y. J. Soh、Z. Wang、Y. Xu、S. R. Dulloor 等，“英特尔 Optane DC 持久内存模块的基本性能测量”，arXiv预印本 arXiv:1903.05714，2019年。[6] DAOS架构。 [在线]。可访问：https://www.alcf.anl.gov/events/daos-next-generation-data-management-exascale [7] M. Weiland、H. Brunst、T. Quintino、N. Johnson、O. Iffrig、S. Smart、C. Herold、A. Bonanni、A. Jackson 和 M. Parsons，“英特尔 Optane DC 持久内存模块的早期评估及其对高性能科学应用的影响”，载于《高性能计算、网络、存储与分析国际会议论文集》，2019年，第1–19页。[8] L. Xiang、X. Zhao、J. Rao、S. Jiang 和 H. Jiang，“英特尔 Optane 持久内存的性能特征：深入考察其内存模块缓冲”，载于《第十七届欧洲计算机系统会议论文集》，2022年，第488–505页。[9] V. Mironov、I. Chernykh、I. Kulikov、A. Moskovsky、E. Epifanovsky 和 A. Kudryavtsev，“使用科学基准测试评估英特尔 Optane DC 内存的性能”，载于《2019年 IEEE/ACM以内存为中心的高性能计算研讨会（MCHPC）》。IEEE，2019年，第1–6页。[10] I. Peng、K. Wu、J. Ren、D. Li 和 M. Gokhale，“揭示基于非易失性内存的内存系统上高性能计算科学应用的性能”，载于《2020年 IEEE 并行与分布式处理国际研讨会（IPDPS）》。IEEE，2020年，第916–925页。[11] R. S. Venkatesh、T. Mason、P. Fernando、G. Eisenhauer 和 A. Gavrilovska，“使用英特尔 Optane 持久内存调度高性能计算工作流”，载于《2021年 IEEE 并行与分布式处理国际研讨会论文集（IPDPSW）》。IEEE，2021年，第56–65页。[12] I. B. Peng、M. B. Gokhale 和 E. W. Green，“英特尔 Optane 字节寻址非易失性存储器的系统评估”，载于《国际内存系统研讨会论文集》，2019年，第304–315页。[13] S. D. Smart、T. Quintino 和 B. Raoult，“面向百亿亿次级数值天气预报与气候的高性能分布式对象存储”，载于《先进科学计算平台会议论文集》，2019年，第1–11页。[14] J. Ejarque、R. M. Badia、L. Albertin、G. Aloisio、E. Baglione、Y. Becerra、S. Boschert、J. R. Berlin、A. D’Anca、D. Elia 等，“为高性能计算、数据分析和人工智能融合实现动态智能工作流”，《未来一代计算机系统》，第134卷，第414–429页，2022年。[15] J. J. Levandoski、D. B. Lomet 和 S. Sengupta，“BW树：面向新型硬件平台的B树”，载于《2013年 IEEE 第29届数据工程国际会议（ICDE）》，2013年，第302–313页。[16] I. Oukid、J. Lasperas、A. Nica、T. Willhalm 和 W. Lehner，“fptree：面向存储级内存的混合SCM-DRAM持久并发B树”，载于《2016年国际数据管理会议论文集》，2016年，第371–386页。[17] S. Chen 和 Q. Jin，“非易失性主存中的持久B+树”，《VLDB基金会会议录》，第8卷，第7期，第786–797页，2015年2月。 [在线]。可访问：https://doi.org/10.14778/2752939.2752947

[18] D. Hwang、W.-H. Kim、Y. Won 和 B. Nam，“字节寻址持久性内存 B+-树中的可持久瞬态不一致性”，载于第16届 USENIX 文件与存储技术会议（FAST 18）。加利福尼亚州奥克兰：USENIX协会，2018年2月，第187–200页。 [在线]。获取地址：https://www.usenix.org/conference/fast18/presentation/hwang [19] Y. Mao、E. Kohler 和 R. T. Morris，“面向快速多核键值存储的缓存优化技巧”，载于第7届 ACM欧洲计算机系统会议论文集，系列：EuroSys ’12。美国纽约州纽约市：计算机协会，2012年，第183–196页。 [ ] 10 1145 2168836 2168855 在线。获取 [20] 地址：https://doi.org/ 。 / 。L. Lersch、X. Hao、I. Oukid、T. Wang 和 T. Willhalm，“持久内存范围索引评估”，VLDB学报，第13卷，第4期，第574–587页，2019年。[21] Y. He、D. Lu、K. Huang 和 T. Wang，“持久内存范围索引评估：第二部分”，arXiv预印本 arXiv:2201.13047，2022年。[22] S. K. Lee、J. Mohan、S. Kashyap、T. Kim 和 V. Chidambaram，“ Recipe：将并发DRAM索引转换为持久内存索引”，载于第27届 ACM操作系统原理研讨会论文集，2019年，第462–477页。[23] X. Zhou、L. Shou、K. Chen、W. Hu 和 G. Chen，“Dptree：持久内存的差分索引”，VLDB学报，第13卷，第4期，第421–434页，2019年。[24] Y. Chen、Y. Lu、K. Fang、Q. Wang 和 J. Shu，“utree：一种尾延迟低的持久性 b+-树”，VLDB学报，第13卷，第12期，第2634–2648页，2020年。[25] M. Weiland 和 B. Homoelle，“高性能和数据密集型计算中字节寻址持久性内存的使用场景”，《计算机科学与技术学报》，第36卷，第1期，第110–122页，2021年。[26] M. Katsaragakis、L. Papadopoulos、C. Baloukas 和 D. Soudris，“异构DRAM/NVM系统中应用程序数据结构优化与放置的内存管理方法”，载于2022年欧洲设计、自动化与测试会议暨展览（DATE）。IEEE，2022年，第748–753页。[27] 持久内存开发套件。 [在线]。获取地址：https: //pmem.io/pmdk/ [28] S. Chen 和 Q. Jin，“非易失性主存中的持久性 b+-树”，VLDB学报，第8卷，第7期，第786–797页，2015年。[29] D. Hwang、W.-H. Kim、Y. Won 和 B. Nam，“ {字节寻址} 持久性 {B+-树}中的可持久瞬态不一致性”，载于第16届 USENIX文件与存储技术会议（FAST 18），2018年，第187–200页。[30] R. M. Yoo、C. J. Hughes、K. Lai 和 R. Rajwar，“英特尔®事务同步扩展在高性能计算中的性能评估”，载于高性能计算、网络、存储与分析国际会议论文集，系列：SC ’13。美国纽约州纽约市：计算机协会， 2013 [ ] 10 1145 2503210 2503232 。在线。 [31] 获取地址：https://doi.org/ 。 / 。[ ] 处理器计数器监控工具。在线。获取地址：https://github.com/ opcm/pcm [32] D. Masouros、S. Xydis 和 D. Soudris，“Rusty：面向现代多租户系统的运行时干扰感知预测监控”，《IEEE并行与分布式系统汇刊》，第32卷，第1期，第184–198页，2020年。[33] PMDK ipmctl。[在线]。获取地址：https://docs.pmem.io/ipmctl-user-guide/ [34] B. F. Cooper、A. Silberstein、E. Tam、R. Ramakrishnan 和 R. Sears，“使用 YCSB 对云服务系统进行基准测试”，载于第1届 ACM云计算研讨会论文集，2010年，第143–154页。[35] J. Boukhobza、S. Rubini、R. Chen 和 Z. Shao，“新兴非易失性存储器：体系结构集成与研究挑战综述”，《ACM电子系统设计自动化汇刊》（ TODAES），第23卷，第2期，第1–32页，2017年。