使用IntelOptaneDC持久内存模块为HPC应用程序设计的DRAM-NVM混合内存体系结构的性能表征

Onkar Patil[1]Latchesar Ionkov[2]，杰森·李[2]弗兰克·穆勒[1]迈克尔·朗[2] [1]北卡罗来纳州立大学
[2]洛斯阿拉莫斯国家实验室opatil@ncsu.edu 穆勒@cs.ncsu.edu
莱昂科夫，杰森利，mlang@lanl.gov

摘要

非易失性、字节寻址存储器(NVM)是由Intel以名为Intel的NVDIMM的形式引入的® 光学™直流PMM。 这个内存模块能够在不需要电源的情况下持久化存储在其中的数据。 这将内存层次结构扩展为混合内存系统，因为来自DRAM的访问延迟和内存带宽的差异，DRAM是主要的字节寻址主存技术。 Optane DC内存模块的容量高达DDR4DRAM模块的8倍，可以将字节地址空间扩展到每个节点6TB。 许多应用程序现在可以在这样一个内存系统的情况下扩大它们的问题大小。 我们评估了这种DRAM-NVM混合内存系统的能力及其对高性能计算(HPC)应用的影响。 我们描述了OptaneDC与DDR4DRAM相比，与STREAM一样的自定义基准，并测量了VPIC、SNAP、ESH和AMG等HPC微型应用在不同配置下的性能。 我们发现，只有Optane的执行在执行时间上比只有DRAM和内存模式的执行慢至少2%至16%的VPIC和最大6倍的ESH。

CCS概念·计算机系统组织→异构（混合）系统；·硬件→内存和密集存储；·计算方法→大规模并行和高性能模拟；

关键词NVM，持久内存，英特尔光电直流，内存分配，混合内存，NUMA，SICM

ACM参考格式：

Onkar Patil[1]Latchesar Ionkov[2]，杰森·李[2]弗兰克·穆勒[1]迈克尔·朗[2]。2019. 使用IntelOptaneDC持久内存模块为HPC应用程序设计的DRAM-NVM混合内存体系结构的性能表征。 *在诉讼中*

记忆系统国际研讨会(MEMSYS‘19)，2019年9月30日至10月3日，华盛顿特区，美国。 ACM，纽约，纽约，美国，16 页数。https://doi.org/10.1145/3357526.3357541

MEMSYS‘19，9月30日-2019年10月3日，华盛顿特区，美国

2019. ACM IS BN978-1-4503-7206-0/19/09.$15.00
https://doi.org/10.1145/3357526.3357541

1 导言

自从计算机被引入和冯·诺依曼体系结构被采用以来，记忆层次一直在不断发展。 今天，半导体存储器以动态随机存取存储器(DRAM)为主，因为它的密度高，成本低[21]. DRAM是不稳定的，容易产生软错误，并且由于保留存储数据所需的不断刷新而更加耗电。 随着处理器时钟频率的增加，引入静态随机访问存储器(SRAM)作为缓存层，以弥补延迟间隙。 随着多级缓存的引入，内存层次不断扩大，高带宽内存被添加，主存储器大小不断增加。 数据持久化，即非波动性，是数据存储的一个特征，是当前内存层次结构的二级层次。 大多数非易失性设备不像DRAM那样在内存总线上，但在延迟方面要远得多。 DRAM和存储中使用的其他技术在访问延迟方面的差异使得在主存层次结构中缩放容量或添加持久性变得很麻烦。

超级计算机是用单个节点构建的，这是
有自己的内存层次结构。 节点通过高速互连连接到其他节点，允许直接存储器访问或远程直接存储器访问。 集群的组合内存大于单个节点的内存，但需要复杂的软件和额外的硬件，而且规模本身也有问题。 例如，橡树岭国家实验室的泰坦[30一台petaflop机器现在已经退役，是2018年11月TOP500名单上最快的超级计算机之一[31]. 每个节点有38GB的DRAM。 作为一个由18,688个节点组成的集群，通过互连连接，泰坦有710TB的DRAM。 然而，系统中使用的DRAM模块的数量太多，使其容易出现软错误和硬故障。 此外，内存是泰坦功耗的主要组成部分之一，在其峰值可达8.2兆瓦。 由于实现exascale内存需求所需的DRAM模块数量较多，构建和操作具有类似内存体系结构的更大机器的成本将增加

很重要。 这也增加了失败的可能性[8]。 因此，具有类似体系结构的exascale机器可能需要硬件创新来解决弹性和功率的挑战。

在过去的十年中，诸如相变存储器(PCM)和自旋转移转矩RAM等存储器技术已经被开发出来，现在被用来制造字节寻址的非易失性存储器设备[1,25]。 虽然它们的速度较慢，但它们的密度比DRAM高。 这种权衡需要进行详细的分析，以评估这些新内存技术的好处。 英特尔是他们的英特尔第一个上市® 基于PCM技术的光电™直流持久存储模块(PMM。 Optane DC通过传统的DIMM插槽直接插入到内存总线中。 它的密度比DRAM高8倍，而且每GB都便宜。 该OptaneDC PMMS可用于扩展具有数据持久性的主存层次结构的容量，也可用作传统的NVM块设备。

使用比DRAM更高密度的记忆
允许Exascale计算机的不同设计点。 可以使用更少的节点来达到更高的聚合内存容量。 节点少意味着组件少，这反过来可以降低构建系统的成本，降低系统的整体功耗，提高弹性。 此外，这些新型内存的数据持久性也有助于开发新的容错机制。

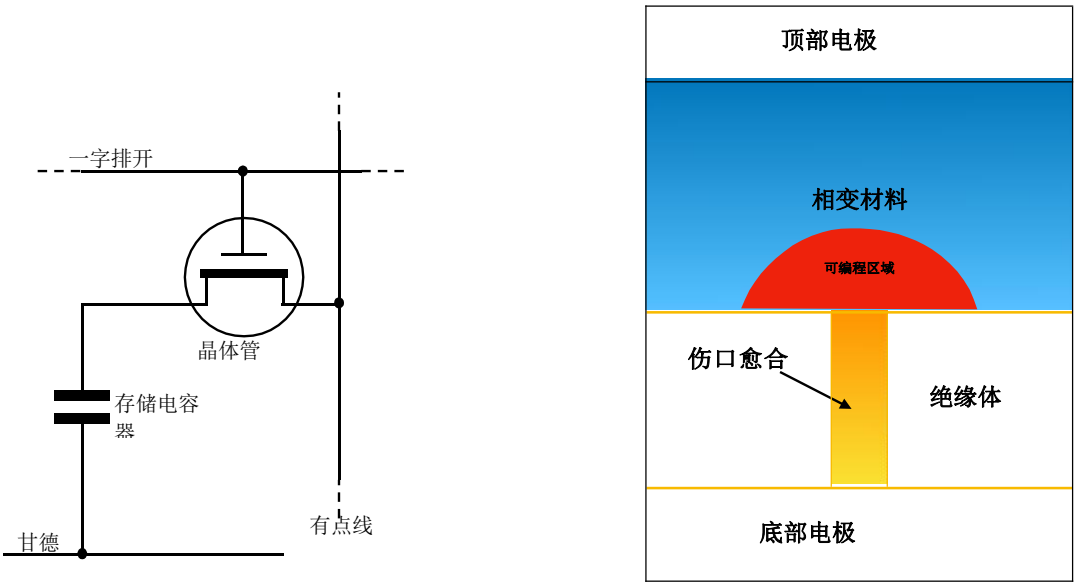
在本文中，我们仔细研究了OptaneDCPMMS，它的底层技术，它的操作，它可以操作的不同模式，并评估它的性能为HPC应用。 我们专注于评估它作为主存的使用，而不是存储系统的一部分。 我们已经通过使用受STREAM启发的自定义基准来表征OptaneDC的性能[18,19]，它具有在HPC应用程序中经常使用的访问流。 我们已经用像VPIC这样的HPC应用程序评估了整个系统的性能[2和代理应用程序，如AMG[39]，lulesh[13和SNAP[29]。 在科室2，我们回顾了与持久记忆系统相关的研究及其评估。 在科室3，我们回顾了非易失性存储器的背景，并在部分4 我们专注于OptaneDC内存体系结构。 在章节中5 还有6，我们评估了OptaneDC的性能。 在科室7，我们提出了未来的潜在工作，并在章节中跟进结论8.

2 相关工作

由于最近推出了OptaneDCPMMS，没有很多以前的工作来评估设备的性能特性。

在[10]，Izraelevitz等人。 评估了OptaneDCPMM的读写特性。 他们使用SPEC2017基准套件评估了OptaneDC在所有可用模式上的性能。 他们发现，与只有DRAM的分配相比，只有NVM的分配应用程序的速度慢了15-61。 他们还比较了不同文件系统和数据库应用程序，如MongoDB和MySQL，分别使用OptaneDC作为持久存储和持久内存。 他们发现，OptaneDC由于比存储设备更低的延迟而提高了文件系统和数据库应用程序的性能。 我们正在评估在不同的HPC应用程序中遇到的不同流的OptaneDC，并重点使用OptaneDC作为相同的扩展地址空间。 Gill等人。[7]使用Optane DC PMM评估共享内存图形框架，如现实世界中的Galois。 他们发现，与现有生产集群上的分布式图形框架相比，OptaneDCPMM为大规模图形分析提供了性能和成本效益。 我们的工作重点是HPC问题，这些问题主要是模板代码和矩阵操作。 Psaropoulos等人。 [24通过交错执行索引连接中的并行工作和使用coroutines的元组重建，隐藏了OptaneDC和DRAM在数据库应用程序中的延迟差异。 它们将NVM和DRAM上的端到端查询运行时间分别提高了1.7倍和2.6倍。 Van Renen等人。 [33]在带宽和延迟方面对Optane DC进行了性能评估，并制定了有效使用OptaneDC的指南和两个调优的I/O原语，即日志写入和块冲洗。 他们的工作主要是基于应用程序直接模式(模式在章节中解释4) 旨在提高文件系统的性能。 Wu等人。 [37]研究了Optane DC早期版本的I/O性能，即NFS和PVFS的3D-Xpoint[14作为文件系统。 它的操作类似于目前的应用程序直接模式在OptaneDC。

在提供软件方面做了很多工作
支持字节寻址的非易失性存储器。 Volos等人。 [35创建了一个简单的接口，用于使用称为Mnemosyne的持久内存进行编程。 它允许程序员分配全局持久和动态数据结构，也允许原语修改数据结构。 Coburn等人。[4实现了一个称为NV堆的轻量级持久对象系统。它提供事务语义，防止错误，并为堆对象提供持久化模型。 Chakrabarti等人。 [3]为基于锁的代码(名为Atlas)提出了一个具有耐久性语义的系统。 它在出现故障时自动保持全局一致状态。 Dulloor等人。 [6]实现了一个POSIX文件系统PMFS，该系统利用持久内存的字节可寻址性来避免面向块的开销



(a) DRAM细胞 **(b)** PC M细胞

图1。 DRAM和PCM的记忆细胞

存储和启用具有内存映射I/O的应用程序的直接持久内存访问。 Yang等人。 [38实现和评估非易失性B的性能+-基于NVDIMM的服务器的NV-Tree和基于它的键值存储。 Shull等人。 [27[参考译文]为Java提出了一个用户友好的NVM框架，以确保崩溃恢复操作的一致存储。

人们提出了将NVM用于HPC系统的各种方法。 Vetter等人。 [34]评估NVM系统用于极端规模HPC的潜力。 他们研究了在HPC中集成NVM的各种持久化设备，还研究了NVM的功能集成。 Kannan等人。 [12]使用NVM作为虚拟内存为HPC应用程序优化检查点，并提供频繁、低开销检查点。 Patil等人。 [23提出了一种新的模板代码编程技术，它保证了在共享的非易失性存储池上对两个硬故障的容错。 Li等人。 [16提出了一种基于NVRAM的容错过程模型，为应用程序容忍系统崩溃提供了一种优雅的方法。 Wang等人。 [36]提出了一种利用NVM作为二次内存分区的新方法，以便应用程序能够显式地分配和操作其中的内存区域。 它有一个允许访问分布式NVM存储系统的库。

3 背景

现代体系结构中的内存层次是复杂而深刻的[21]. 寄存器内存最接近处理器，处理器用于加载和存储操作符、操作数和指令。 它是使用触发器或数组实现的SRAM细胞。 它是最快的内存，非常昂贵，消耗了大量的芯片空间和功率。 大多数现代CPU有16到32个寄存器，每个寄存器可以容纳32或64位。 寄存器存储器的访问时间小于1ns。 由于计算机程序比寄存器需要更多的内存，我们使用比寄存器更便宜和更高密度的内存作为我们的主存储器。

主存储器由DRAM单元阵列组成，如图所示。 1a。 它是一种基于半导体的存储技术，它将一位数据存储在集成电路中的电容器中。 它是一个矩形的单元阵列，存储电荷，由每个数据位的电容器和晶体管组成。 细胞数定义DRAM芯片的容量。 有正负位线连接一列中的所有单元格。 位线之间的一对交叉连接的逆变器，称为感觉放大器，用于稳定存储在单元中的电荷。 由于电池中的电荷泄漏，DRAM必须不断刷新以保持其状态[9]. 的JEDEC标准[11]指定每个行必须每64ms或更少刷新一次。 由于这种不断需要刷新电荷，这种记忆使用了更多的能量。 使用DDR4协议访问DRAM的时间约为50-100ns。 对DRAM的访问比寄存器访问时间慢10-15倍，这可能导致许多CPU停滞。 处理器使用SRAM缓存作为缓冲区来隐藏这种延迟。 SRAM使用锁存器存储每个位。 它们是不稳定的，当内存没有动力时，它们就会失去状态。 它被称为静态，因为它不需要刷新。 现代内存体系结构利用多级缓存来减少主存和处理器之间的有效延迟。 的

对SRAM缓存的访问时间取决于缓存的级别，平均为1-10ns。 在芯片空间和功率方面，SRAM比DRAM更昂贵。

这种记忆层次在几十年的时间里已经演变并变得更加复杂。 在过去的20年里，由于DRAM容量和频率的不断扩大，它的规模也变得更快。 但在同一时期内，DRAM缩放比核心计数缩放慢了大约33。 此外，由于内存单元的数量增加和较高的刷新率，DRAM的能耗增加了[17, 20, 26]. 尽管DRAM容量增加，但由于密度较高，这些记忆变得不那么可靠[8]. 在HPC、机器学习、图形分析和其他领域中的极端规模问题可能耗尽节点内存储容量和处理。 [34]. 使用NVM作为DRAM的补充，以增加计算节点中主内存的大小，已被认为是一种可行的选择[15]. 在所有的NVM技术中，PCM在工程方面发展得最好[15]. PCM是电阻存储器，DRAM是电荷存储器。 如图所示。 1b，PCM有位线，它是通过加热器连接到相变材料的金属。 当电流脉冲通过位线时，在相变材料中设置一个相位，并存储在那里，直到另一个电流脉冲通过为止。 通过访问线检测材料的电阻来读取相位。 相变材料在环境温度下保持其相超过10年[15]。 此属性使PCM具有非挥发性。预计PCM将缩小到9nm，而小于40-35nm的DRAM具有挑战性[20].

然而，PCM有其自身的挑战和不足。
由于改变相变材料所需的热活化，它具有比DRAM更高的写入延迟。由于位线和相变材料之间的接触发生热膨胀和收缩，PCM也受到磨损。PCM存储单元的可写性约为10^8这意味着需要频繁的设备替换，这可以增加成本[15]. 尽管存在这些缺点，PCM提供了缩放匹配核心计数缩放所需的主存容量的能力。 英特尔的OptaneDCPMMS是基于PCM技术的。

4 建筑

我们用于实验的系统是由英特尔公司提供的一个节点。 如表所述1，这个节点有两个CPU插座，配备了英特尔的Xeon® 8260L（代号为级联湖）。 每个芯片有24个核心，时钟频率为2.4GHz。 每个核心有2个处理单元，总共96个CPU。 每个核心都有32KB私有L1指令缓存，32KB私有数据缓存，以及

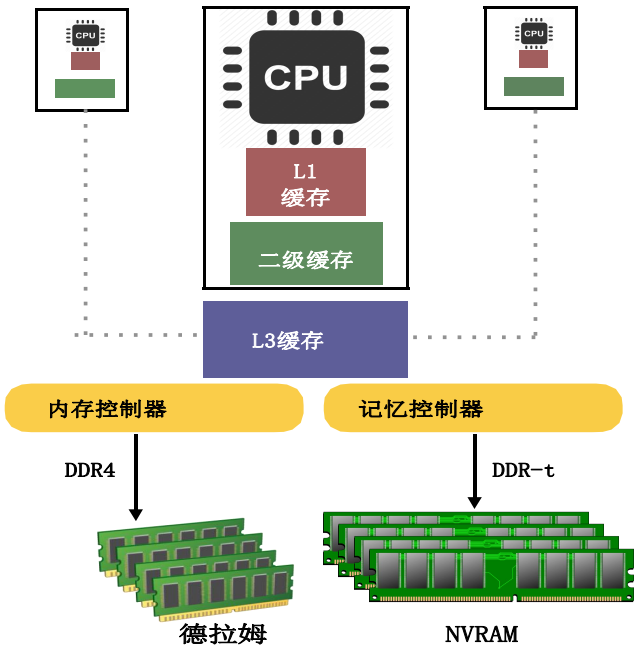


图2。 英特尔的OptaneDC节点的内存体系结构

表1. 实验平台

规格	选择节点
模型名称架构	英特尔(R)Xeon(R)8260L@2.40G Hz
	x86_64
中央处理器	96
口袋	2
每个插座都有铜芯	24
NUMA节点	4
L1D缓存	32kb
L1i缓存	32kb
二级缓存	1mb
L3缓存	35.3mb
记忆控制器	4
频道/控制器	6
DIMM协议	DDR4
DRAM大小	192GB
NVDIMM协议	DDR-t
NVRAM大小	1.5TB
操作系统	Fedora27

一个私有的1MBL2缓存。 所有核心之间共享一个36MB的L3缓存。 每个插座有12个DIMM插槽。 其中6个插槽被16个GB DDR4DRAM模块占用，其余6个插槽被128个GB光电直流模块占用。 这总计高达192GB的DRAM和1.5TB的非易失性存储器。 该节点共有4个内存控制器。 其中两个存储器控制器分别连接到6个DRAMDIMM，另两个称为IMC的存储器控制器分别连接到6个NVDIMM。

如图所示2，处理器与OptaneDCDIMM的通信方式与DRAM不同。 对于DRAM，它通过常规使用标准的DDR4协议

内存控制器，而对于OptaneDC，它使用DDR-T协议通过I-内存控制器(IMC)。 使用DDR4协议的专有附加协议，OptaneDC实现异步命令/数据定时和可变延迟内存事务。 为了与IMC通信，OptaneDCPMM中的模块控制器使用请求/授予方案。 Optane DC驻留的数据总线的方向和定时由处理器控制。 处理器根据请求向OptaneDC内存控制器发送命令包。 模块使用256字节缓存线访问粒度，大于DRAM中使用的64字节缓存线访问粒度。 [10]. 英特尔的异步DRAM刷新(ADR)保证CPU存储达到它，将在电源故障中生存。 商店在不到100µs内被冲洗到NVDIMM，这是暂停时间。 iMC落在ADR域中，但是缓存没有。 因此，存储只有在到达iMC后才会持久化，它使用72位数据总线，并以缓存线粒度传输CPU负载和存储的数据。 Optane DC有一个on-DIMM Apache Pass控制器，它处理内存访问请求和NVDIMM上所需的处理。 在DIMM控制器内部转换所有访问请求的地址，用于磨损流平和坏块管理。 它在DIMM上维护一个地址间接表，将DIMM的物理地址转换为内部设备地址。 表也备份在DRAM上。 访问OptaneDC上的数据发生在翻译之后。 控制器将64个字节的负载/存储转换为256个字节的访问，因为OptaneDC的高速缓存线访问粒度较高，从而导致写入放大[10].

光电直流工作在三种不同的模式。 与
一个小的Linux内核修改，我们已经配置了OptaneDC以第四种模式操作。 配置描述如下。

4.1 记忆模式

在内存模式下，OptaneDC模块充当易失性主存。 DRAM作为Optane DC的直接映射缓存，块大小为4KB，由CPU的内存控制器管理。 DRAM不再是直接访问的，而是允许缓存点击速度与DRAM访问速度一样快。 然而，缓存丢失可能需要只要DRAM缓存丢失加上OptaneDC访问。

4.2 应用程序直接模式

在AppDirect模式中，OptaneDC模块充当与主存储器分离的持久存储设备。 使用DRAM作为主存储器。 但是，Optane DC DIMM通过内核中创建的块设备条目使用。 一旦在每个设备上安装了文件系统，OptaneDC模块就被用作文件系统，其访问时间明显短于常规存储设备。

4.3 混合模式

Optane DC模块也可以进行分区，将部分内存用于持久内存，而另一部分用作易失性主存。 DRAM仍然被用作主存储器的缓存，而不是像在整个应用程序直接模式下那样被公开。

4.4 DRAM-NVM混合模式（平面模式）

在以前的配置下，不可能同时在统一的字节寻址地址空间下访问DRAM和OptaneDC。 在[10]，Izraelevitz等人。 运行实验，他们的OptaneDC模块不被系统的DRAM缓存，通过修改Linux内核来识别OptaneDC模块为RAM，而不是持久内存。 我们将这些更改应用于节点的内核，并将DIMM设置为AppDirect模式，允许我们除了DRAMNUMA节点之外，还可以在NUMA节点上看到OptaneDC模块，这将导致DRAM容量加上OptaneDC容量的合并主内存，而不仅仅是一个或另一个。

表2光电直流工作模式

操作模式	功能性
记忆模式	光学DC PMMS具有挥发性， 字节寻址主存储器。 DRAM作为Optane的缓存 直流且用户不可见
应用程序直接模式	光学直流PMMS作为持久性 与主存储器层次结构分开的存储。 由安装在上面的文件系统管理。 DRAM作为主要存储器
混合模式	部分OptaneDCPMM可以 用作主存储器，其余部分可用作持久存储。 DRAM行动 作为OptaneDC的缓存
平模式	DRAM和Optane DC PMMS 是相同地址空间的一部分 可以用作堆内存

5 实验

我们的目的是评估OptaneDC作为HPC系统中主存储器的地址空间扩展器。 我们使用一个HPC应用程序(VPIC)、三个HPC代理应用程序(AMG、ESH和SNAP)和一个自定义基准来评估OptaneDC的性能。 我们修改了这些

应用程序[1] 因此，我们可以在DRAM或OptaneDC上分配所有数据。 然后，我们比较了我们收集的两种配置的统计数据。 我们使用类似STREAM的自定义基准进行了初步性能表征，该基准评估了在HPC应用程序中遇到的不同类型内核的性能。 为使用OpenMP并行化的内核中使用的每个流收集内存带宽信息[5]. 基准中使用的流代表了在HPC应用程序中发现的大多数流。 我们关注数据结构的不同访问模式，如顺序访问、跨行访问和随机访问。 我们还收集矩阵访问和操作的带宽编号，例如行主访问和模板操作。 我们有一个测试，在那里我们绕过L3缓存，通过访问按适合L3缓存的元素数分隔的元素。

我们的实验运行在本节中描述的OptaneDC节点上4 还有桌子1. 我们在内存模式和DRAM-NVM混合模式（平坦模式）中设置了OptaneDC模块，并比较了每个模式的所有应用程序的性能。 在平面模式下，我们在NVM和DRAM上为不同的运行分配内存。 在内存模式下，内存只在NVM上分配，因为DRAM被用作NVM的缓存。

我们对所有HPC迷你应用程序进行了强和弱的缩放，并测量了总执行时间、内存带宽、功耗、最后一级缓存丢失和每秒双精度浮点操作。 我们用了Likwid[32收集性能计数器。 对于我们的自定义基准，我们只收集了我们测试的不同内核的内存带宽。

考虑到我们只有一个节点上有OptaneDCPMMS，我们的实验不是在大量的进程或内存大小上进行的。 我们确保我们的问题大小足够大，不适合最后级别的缓存，这样我们就可以公平地描述不同内存的性能。 我们的问题大小在一个小/中的范围内，由作者推荐的迷你应用程序。 我们没有将流程的数量扩大到48个以上，即处理单元数量的一半，以避免资源的过度订阅。 这样做是为了从硬件性能计数器中获得正确的性能编号。 对于自定义基准，我们对每个标准偏差高达8%的内核平均带宽测量超过10次运行%。 对于HPC迷你应用程序，我们平均所有测量超过4次运行，标准偏差为11%的执行时间。 我们描述了我们在下面的实验中使用的应用。

[1]为了使应用程序只在OptaneDC或DRAM上分配数据，我们修改了应用程序，以使用复杂内存的简单接口(SICM)[28库，一个NUMA感知的异构内存竞技场分配器。

5.1 安格

AMG是一个并行的代数多网格求解器，用于非结构化网格上的问题[39]. 它是在劳伦斯利弗莫尔国家实验室(LLNL)开发的)。 它是一个SPMD代码，在MPI任务中使用MPI和OpenMP线程。 AMG是一个高度同步的代码。 通信和计算模式表现出许多并行科学代码共同的表面到体积关系。 我们在带有27点模板的立方体上使用默认的拉普拉斯类型问题。

5.2 卢勒什

卢勒什[13是一个高度简化的应用程序，硬编码只解决一个简单的Sedov爆炸问题与解析答案。 它是基于C++的应用程序。 它是在LLNL开发的，作为exascale计算的共同设计工作的一部分。 通过将空间问题域划分为一个由网格定义的体积元素集合来离散逼近流体力学方程。 它使用MPI和OpenMP进行并行化，也是内存绑定。

5.3 vpic

矢量粒子内盒(VPIC)[2]是在洛斯阿拉莫斯国家实验室(LANL)开发的模拟代码)。 这是一个应用，在1到3维建模动力学等离子体。 它使用MPI和OpenMP进行并行。 该代码由同时计算多个数据流并对整个数据结构进行操作的内核组成。 数据结构基于输入甲板进行缩放，从而使VPIC内存绑定。

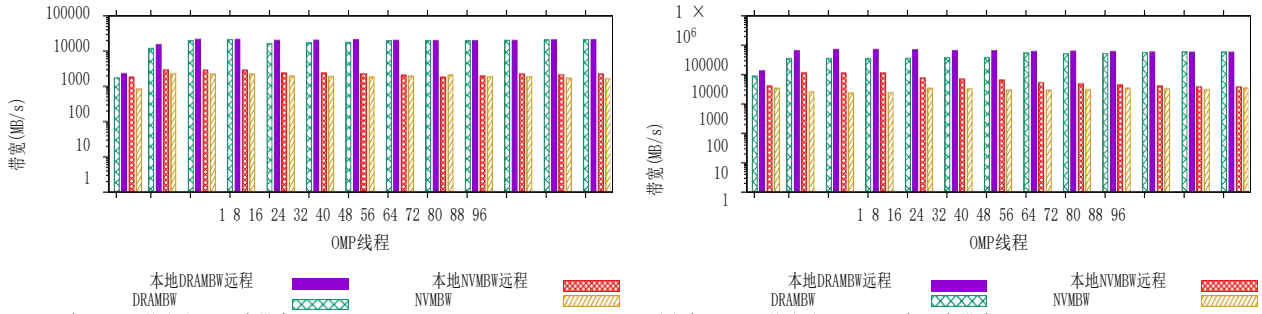
5.4 啪

折断[29]基于LANL的PARTISN代码。 SNAP模拟PARTISN的计算工作量、内存需求和通信模式。 它所解决的方程是使用相同数量的操作，使用相同的数据布局，并以大致相同的顺序加载数组的元素。 SNAP使用MPI对HPC进行缩放。 我们使用SNAP-C代码。 它也是一个内存绑定的应用程序，但更受带宽而不是延迟的约束。

6 结果

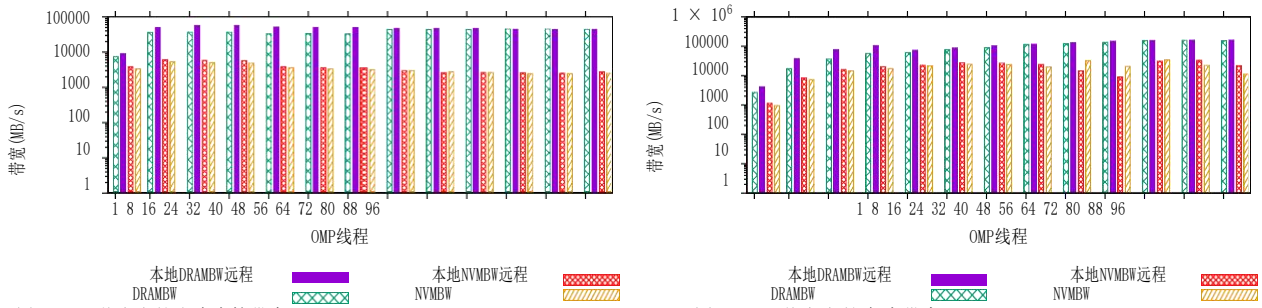
6.1 不同流在OptaneDC上的性能评估

与DRAM相比，我们评估了在OptaneDC上执行的各种流的性能。 对于不同的运行，我们通过将Open MP线程从1增加到96来对流执行强缩放。 我们使用numactl-C将线程固定到特定的处理单元，然后在每个NUMA节点上分配流，以评估NUMA距离对内存带宽的影响。 我们



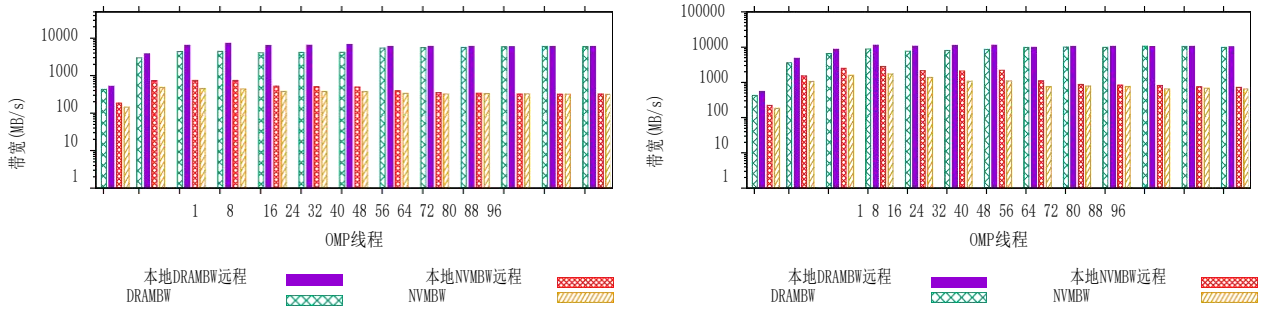
(a) 在Optane节点上只写流带宽 (b)在Optane节点上写入1+4读取流带宽

图3只写和1写+4读顺序访问流的带宽测量



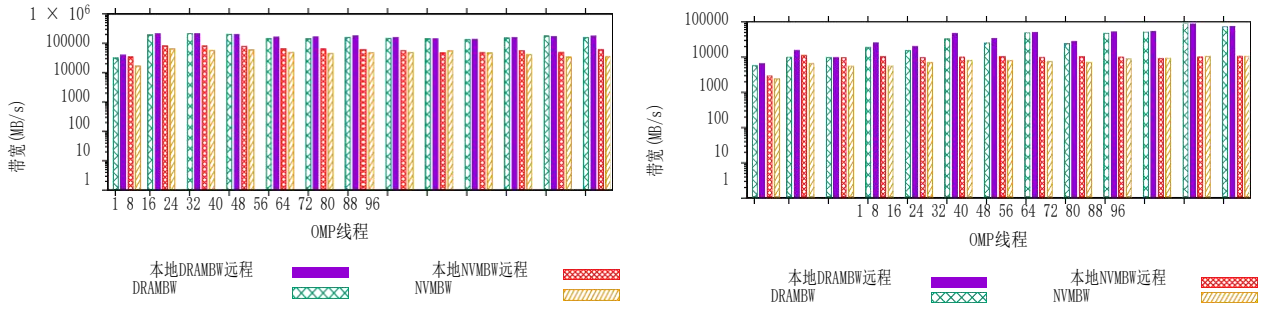
(a)Optane节点上的多流内核带宽 (b)Optane节点上的串流带宽

图4. 多个1写+4读顺序访问和固定步长访问流的带宽测量



(a)Optane节点上的随机流带宽 (b)LLC绕过Optane节点上的流带宽

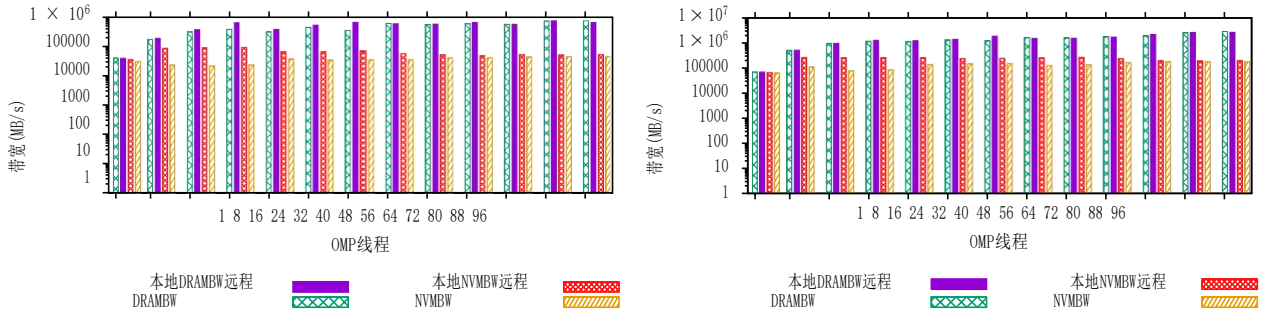
图5. 随机和LLC旁路步长接入流的带宽测量



(a) 行Optane节点上的主矩阵流带宽 **(b)** 列在Optane节点上的主矩阵流带宽

图6行主和列主矩阵访问流的带宽测量

收集了所有处理单元及其本地和远程NUMA节点组合的有效带宽，并对10次运行的带宽结果进行了平均，标准差为7%。 每个流或数据结构实验中使用的尺寸为1GB。 所有的图都有Y轴上的带宽，这是在日志尺度上描述的，以及X轴上的线程数。



(a)在Optane节点上行+列主矩阵流带宽 (b)Optane节点上的9个单元模板流带宽

图7. 行柱矩阵和9点模板流的带宽测量

无花果。 3a描述了只写流强缩放的结果。 本地NUMA节点的平均带宽在48个线程处达到峰值，然后再进行平台化。 远程NUMA节点的带宽也随着开放MP线程的数量和24个线程的峰值而增加。 这种影响是由于过度订阅超过48个线程的资源，从而导致内存控制器队列溢出。 这可能导致负载和存储由于背压而序列化，并取消银行并行性的好处。 我们观察到，NUMA距离影响NVM的带宽高达22%，DRAM的带宽高达16%，以防只写流的强缩放。 无花果。 3b显示具有单个写入和四个读取流的内核的结果。 与只写流的效果相同，但每次运行的相对带宽超过只写流的带宽3倍。 对于DRAM和NVM都观察到了这种效果，但DRAM在过度订阅之前比NVM获得了8倍高的带宽。 DRAM和NVM带宽之间的差异恶化到超过48个线程，它们开始稳定。 这种顺序访问的流主要用于HPC应用程序的初始化或问题生成阶段。 上述结果表明，在此阶段，局部DRAM节点的利用是至关重要的，而不是过度订阅来计算资源。

无花果。 4a为具有多个单个的内核提供结果
写加4读流。 性能类似于观察到的单写和四读流，但有效带宽略低。 DRAM的带宽比单写和四读流小30%，NVM的带宽小50。 这种流不需要大量的并行性来实现最大带宽，但是内存设备的访问延迟会影响性能。 无花果。 4b描述具有固定步长访问的内核用于增加OpenMP线程的结果。 步幅大于缓存线大小。 对于这个流，NUMA距离对内存带宽没有影响，除了DRAM的少于24个线程。 NVM的规模与以前的流相似，但实现了更高的内存带宽。 DRAM的内存带宽随着线程数量的增加而不断增加，直到达到48个峰值用于NVM的线程。 这表明NVM带宽可能受到核心计数的限制，而不管访问模式如何。 这个流达到了DRAM的只写流的带宽8倍，NVM的带宽10倍。 无花果。 5a显示随机访问的单写加4读流的结果，这达到了所有流的最低有效带宽。 随机访问由间接数组决定，该数组由Rand()函数初始化，该函数生成访问索引的顺序。 本地DRAM和NVM节点的内存带宽高达40%，直到48个线程停止，并且不受NUMA距离的影响。 之所以观察到这种效应，是因为不允许HW预取器利用任何时间局部性。 因此，有效带宽是如此之低。 无花果。 5b说明了单个写加4读流的结果，强制绕过每个访问的L3缓存。 在这个流中，我们观察到DRAM实现的带宽比NVM高5倍，直到48个线程。 带宽仍然是所有NUMA节点的平台。 这一结果表明，如果我们将缓存从图片中取出，OptaneDC实现的有效带宽不会受到DRAM的影响，尽管访问延迟较高。 这种具有不同访问模式的流在HPC应用程序的计算阶段是常见的。 必须在程序的每个阶段识别每个线性流的访问模式，并将该流放置在给定订阅的计算资源量的最有效内存节点中。

无花果。 6a描述访问单个流的结果
按行主要顺序写和两个读取矩阵。 缩放模式类似于只写流，但有效带宽几乎是DRAM带宽的10倍，而NVM的带宽高达40倍。 这种高带宽可以观察到，因为大的缓存大小和预取，这利用了空间局部性。 同样，这种流在HPC应用程序的初始化阶段是一个常见的事件，它可以从放置在本地DRAM内存中获益。 无花果。 6b评估一个流，以列的主要顺序访问一个写入和两个读取矩阵。 在这里，

随着线程的增加，NVM带宽保持稳定，对本地NVM节点略有优势。 然而，对于DRAM，NUMA距离没有很大的差异，带宽随着线程数量的增加而不断增加。 对于DRAM，带宽比大多数线程的行主流低3倍，NVM的带宽低8倍，除了2个最高的线程计数外，其中DRAM带宽高达86GB/s。 无花果。 7a显示访问单个写入加两个读取矩阵流的内核的结果，除列主序列中的最后一个读取流外，所有这些都是行表示的。 对于DRAM和NVM，它实现了比行主流更高的4倍内存带宽。 由于空间局部性和缓存中的预取，它实现了如此高的带宽。 缩放模式类似于行主访问流，但对两个存储器实现了更高的带宽。 较低的线程计数为本地NUMA节点提供了优势，但超过48个线程，没有差异。 无花果。 7b描绘了一个9点模板内核的结果，它的尺度类似图。 7a但在大约40倍的带宽下，所有线程计数和内存。 由于缓存中存在大量的空间和时间局部性，该流实现了所有流的最高带宽。 观察到的带宽实际上是缓存的带宽。 这种矩阵流发生在HPC应用程序的计算阶段。 虽然NUMA距离确实影响这些流的有效带宽，但使用的内存设备随着线程数量的增加而对带宽产生显著影响。 此外，有效地使用缓存局部性有助于实现两个内存设备的更高性能。

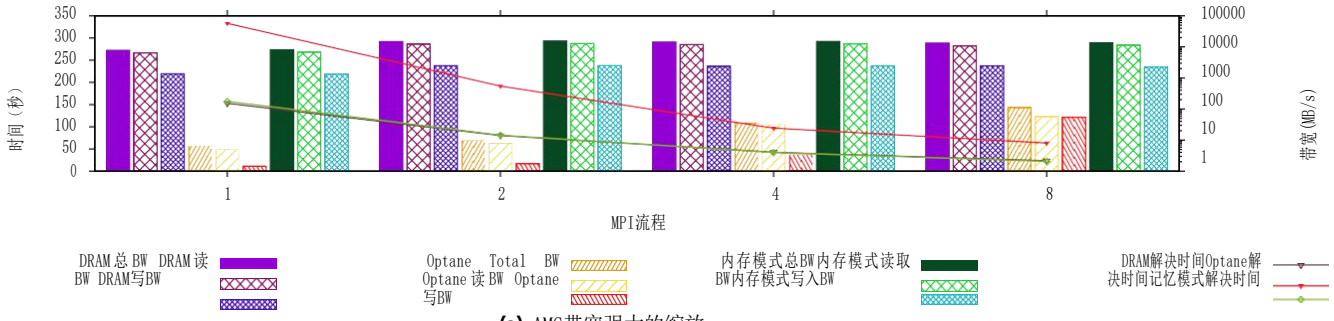
考虑到所有的结果，我们可以推断出
较高的延迟OptaneDC和缺乏最佳的缓存支持导致它不能执行以及DRAM。 我们观察到，通过有效的缓存和预取，OptaneDC可以提供比我们的评估中观察到的更好的性能。 然而，这些结果给出了一个公平的想法，哪些工作负载可以从NVM中受益，并通过使用NVM代替DRAM来量化性能影响。

6.2 HPC基准评估

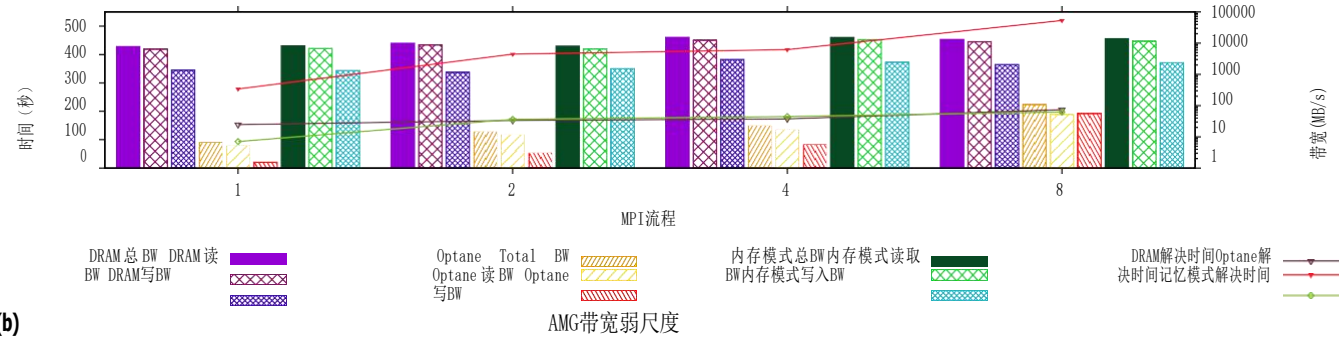
对于基准，我们将应用程序带宽和执行时间度量一起绘制在一个图中，以观察它们之间的相关性。 同样，我们将能耗和执行时间测量一起绘制。 我们还绘制了循环/指令(CPI)和L3错过比在一起。 我们绘制了这些图，用于强和弱的缩放实验。 对于带宽和执行时间图，我们以秒为单位在左侧y轴上绘制执行时间，并描述为行。 带宽以兆字节/秒(MB/s)绘制在右侧y轴上，并描述为条形图。 对于能耗和执行时间图，我们再次绘制我们的执行时间
左边的y轴作为线。 能量在Joules(J)中的右侧y轴上绘制为条形图。 对于CPI和L3错过比图，我们将左侧y轴上的CPI绘制成线。 在右侧y轴上绘制了L3漏失比作为条形图。 两个CPI和L3错过比率都没有单位。 带宽是在对数尺度上绘制的，而所有其他测量都是在线性尺度上绘制的。 x轴描述了给定执行的MPI进程的数量。

6.2.1 AMG结果

无花果。 8a还有图。 8b分别描述了AMG的强标度和弱标度图。 对于强缩放，我们使用MPI将进程从1缩放到8，并通过将每个处理器的大小从256缩小到128来保持数据大小不变。对于弱缩放，我们再次将进程从1缩放到8，并将每个处理器的大小保持在256，就像我们按比例缩放数据大小一样。 我们观察到，只执行Optane的内存带宽比只执行DRAM和内存模式执行低2到3个数量级。 这导致超过2倍以上的执行时间，只有Optane执行。 这一结果是观察到的强弱缩放情况。 对于仅用于DRAM和内存模式执行的所有进程的强和弱缩放，观察到的带宽保持相当恒定，但仅用于Optane执行的带宽则上升。 只有Optane执行的带宽较低是OptaneDC访问延迟较高的结果。 内存模式执行与只执行DRAM的性能相匹配，因为它使用DRAM作为缓存。 这个实验的问题大小足够小，适合DRAM。 因此，只有DRAM和内存模式执行的性能差别很小。 无花果。 9a 还有图。 9b 描述AMG的三次执行的能耗和执行时间，用于强和弱的缩放。 我们观察到只有Optane执行的能耗比只有DRAM和内存模式执行的能耗高2倍。这是由于其较高的执行时间，即使光执行消耗的功率低于其他执行。 无花果。 10a还有图。 10b描述L3缓存丢失和循环/指令(CPI)的强弱缩放AMG的所有3个执行。 在强缩放中，我们观察到只有Optane执行的CPI对于低数量的进程更高。 对于更多的进程，它们几乎等于其他2次执行。 然而，L3缓存缺失随着进程的数量迅速增加。 这也解释了只执行Optane和其他2次执行的执行时间的差异。 在只有Optane的执行中，L3缓存丢失的增加在弱缩放下也被观察到，但是对于其他2个执行，CPI总是高于CPI。 AMG是一个内存绑定应用程序，受到内存访问速度的严重影响。 因此，只有Optane的处决

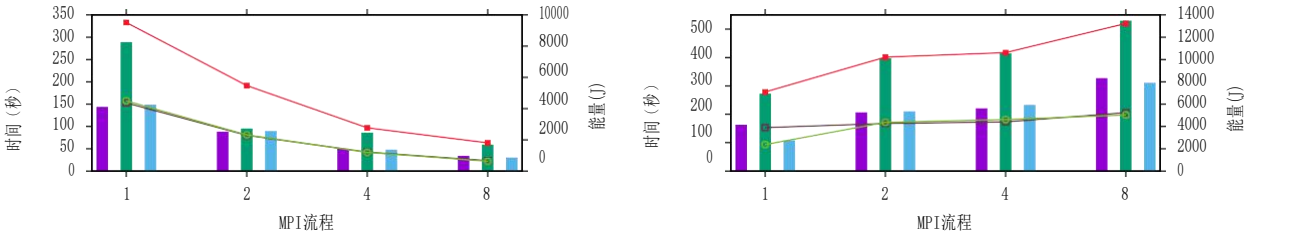


(a) AMG带宽强大的缩放



(b) AMG带宽弱尺度

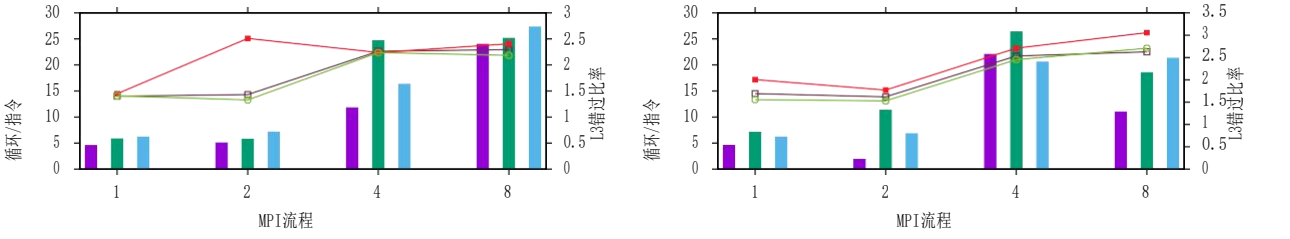
图8AMG的带宽测量



(a)AMG记忆能量消耗强弱

(b)AMG内存能耗弱缩放

图9. 内存能耗为AMG



(a)AMG L3漏计率和CPI强尺度

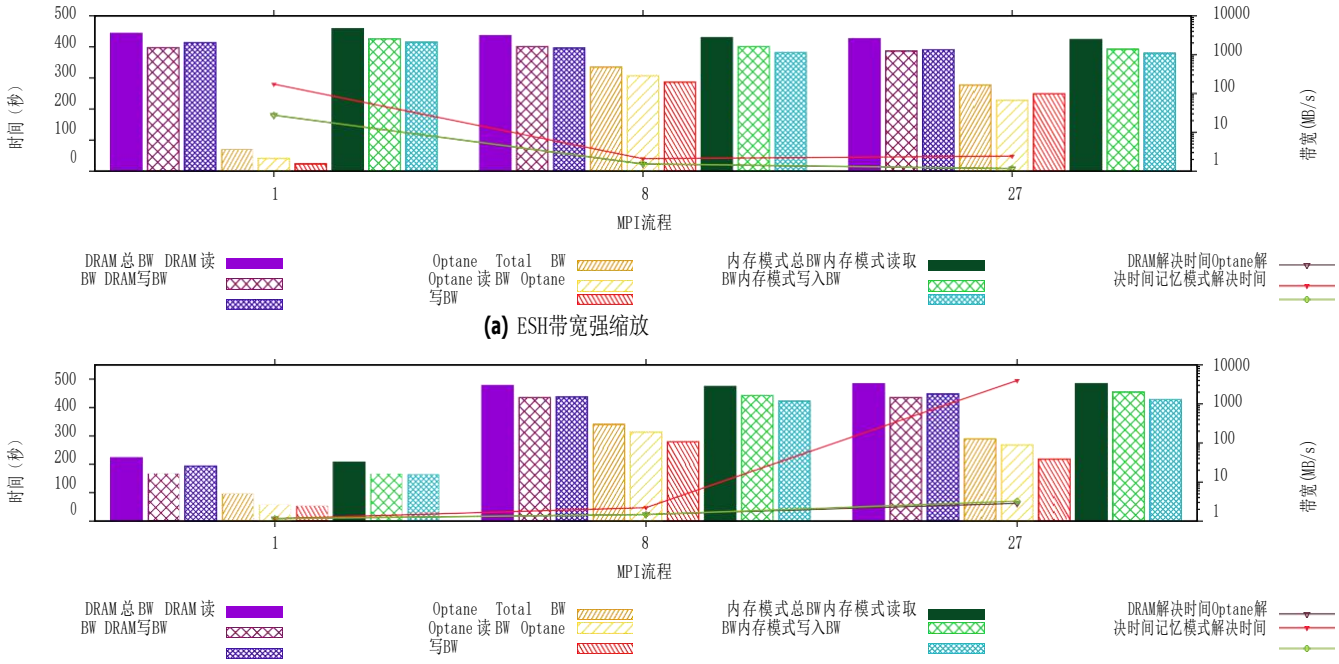
(b)AMG L3漏计率和CPI弱尺度

图10AMG的L3漏计率和CPI

在执行时间和能耗方面遭受严重的性能退化。 这种需要更快访问速度的应用程序将受到仅NVM方法的影响。

6.2.2 结果

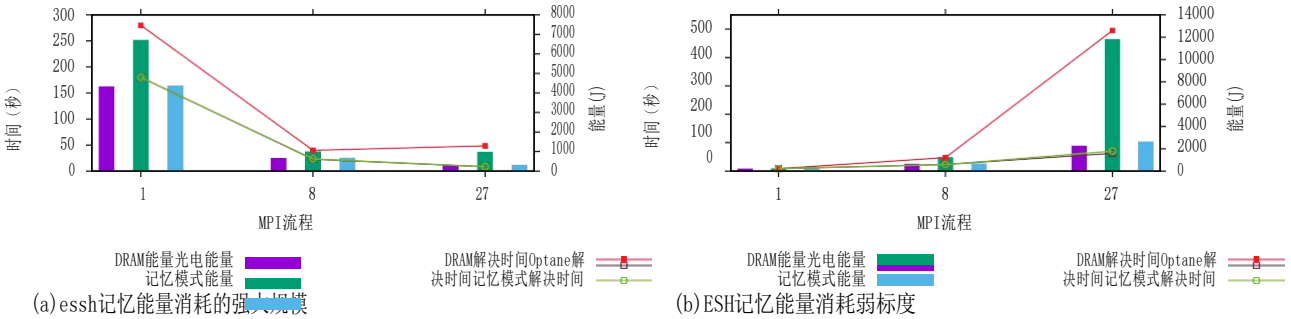
无花果。 11a还有图。 11b分别描述了ESH的强标度和弱标度的图。 我们使用MPI将处理器数量从1增加到27，因为ESH只接受自然数的立方体作为有效配置。 为了



(a) ESH带宽强缩放

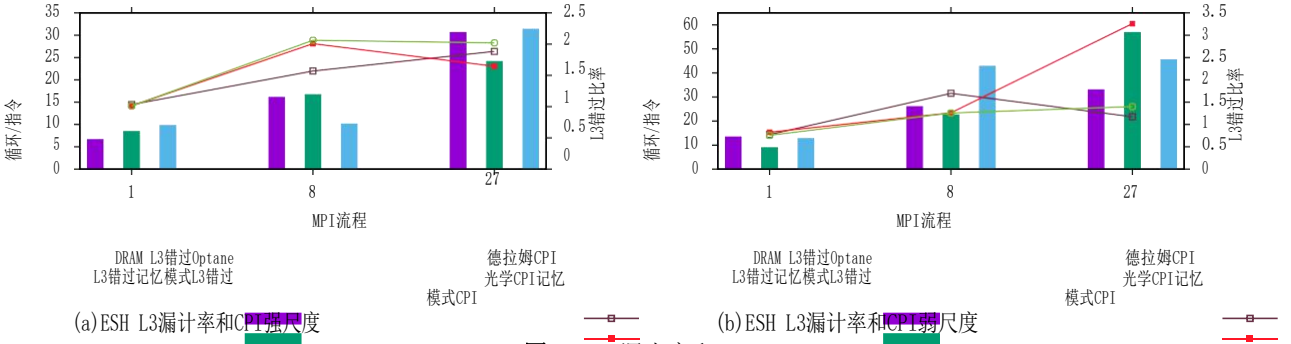
(b) 带宽弱尺度

图11带宽测量



(a)essh记忆能量消耗的强大规模 (b)ESH记忆能量消耗弱标度

图12。内存能耗为ESH



(a)ESH L3漏计率和CPI强尺度 (b)ESH L3漏计率和CPI弱尺度

图13。L3漏失率和CPI

强缩放，我们保持问题大小在12.5万个数据点不变，并增加处理器的数量。 我们观察到，当ESH只在平模式的OptaneDC上运行时，它的执行时间比OptaneDC高50

过程和8个过程。 对于27个进程，执行时间大约高出6倍。 观察到这种效应是因为内存带宽几乎比仅DRAM和内存模式低一个数量级

只有DRAM和内存模式配置的单一

处决。 执行时间的差异较多

由于带宽较低，27个进程在弱缩放中被放大。 在弱缩放中，我们将每个进程的数据点数目保持在15,625。 只有Optane执行的低带宽是由于OptaneDC的高访问延迟所致。 由于内存模式中的DRAM缓存，我们在内存模式中没有观察到这种效果。 因此，只有DRAM和内存模式执行的执行时间没有区别，因为访问延迟是相同的。 在平面模式下，英特尔的ADR正在工作，以保证数据的持久性，这可能会阻碍性能，写放大也可能增加访问延迟。 弱缩放中的内存带宽类似于强缩放，除了强缩放中的单个进程执行外，其中OptaneDC具有2个数量级的低带宽。 无花果。 12a还有12b 分别描述了在强缩放和弱缩放的每种模式下的内存能耗。 我们观察到，OptaneDC的能耗比仅DRAM和内存模式执行高出60。 这与执行时间直接相关，因为OptaneDC的功耗比DRAM低30。 对于较小的问题大小，在较少的线程弱缩放下，OptaneDC的能耗类似于其他执行。 需要在容量、问题大小和性能之间进行权衡，以将应用程序的执行控制在所需的能源预算之内。 无花果。 13a还有图。 13b分别描述了L3缓存漏失比和CPI的强和弱标度。 对于强缩放，L3缓存错过随着进程数量的增加而增加，但它们在27个节点上仅用于Optane执行时较低。 在弱缩放中，CPI比仅DRAM高3倍，L3缓存丢失明显更高。 这些都增加了执行时间，解释了ESH的弱缩放在执行时间上的差异。像ESH这样的应用程序的性能取决于内存带宽。 这些应用程序在运行较小的问题大小和较少的线程时，可以减少NVM的能耗。

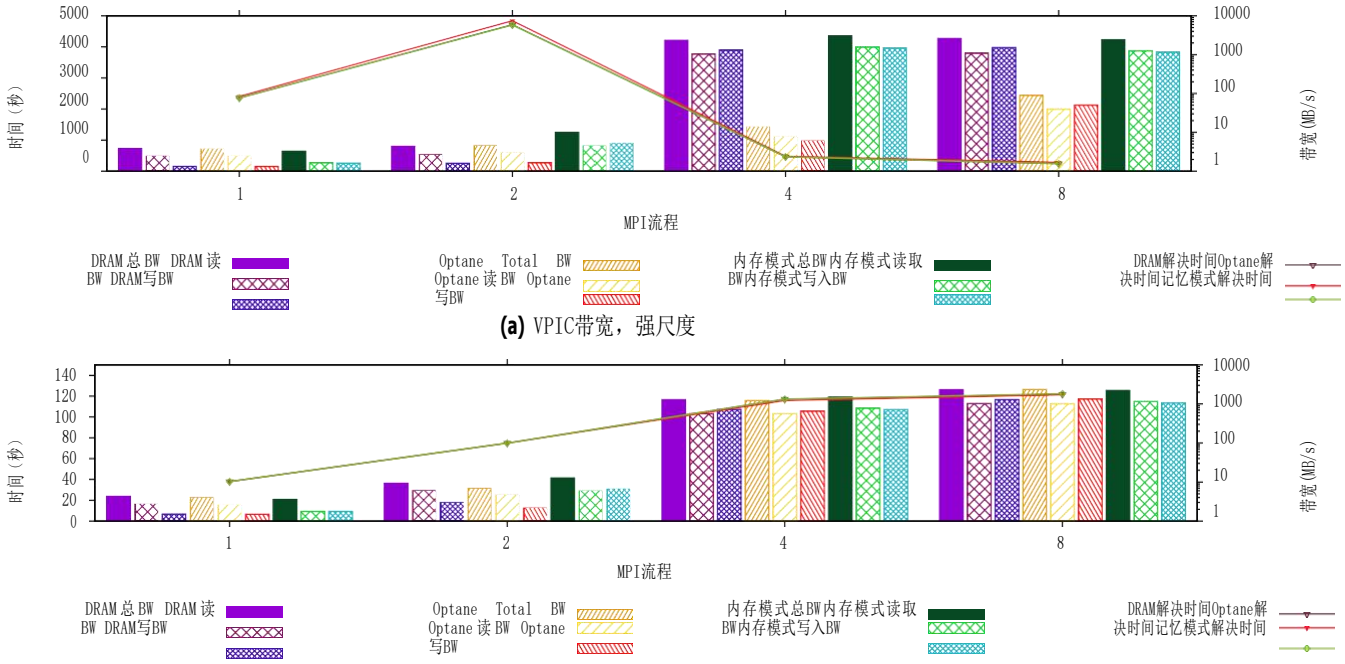
6.2.3 VPIC结果

无花果。 14a还有图。 14b描述VPIC强、弱缩放的执行时间和内存带宽。 我们使用由我们的实验基准作者提供的“LPI”输入甲板。 对于强缩放，我们使用MPI将进程数量从1增加到8，并通过将“nppc”值从2048更改为256来保持问题大小不变。 输入甲板中的“nppc”变量决定了等离子体中每个物种的粒子/细胞数量。 我们观察到，只有NVM执行VPIC比仅DRAM和内存模式执行慢2到16。 对于弱缩放，我们通过将‘nppc’值保持在512来保持每个进程的问题大小相同。 这种减速是由于观察到的带宽较低所致

只有选择执行。 在强缩放的情况下，对于更多的进程，光单存储器带宽至少比仅DRAM和内存模式带宽低一个数量级。 对于弱尺度，仅Optane执行的内存带宽类似于仅DRAM执行和内存模式执行，因此执行时间也没有差异。 无花果。 15a 还有图。 15b 描述了VPIC强尺度和弱尺度的内存能耗。 只使用Optane的执行的能耗保持不变，VPIC的比例很强，类似于DRAM的能耗。然而，在弱尺度下，仅Optane执行的能耗上升速度比其他2次执行的能耗上升速度慢。 由于在弱缩放下所有执行的执行时间都是相似的，所以所有三个执行所消耗的能量也是相似的。 无花果。 16a还有图。 16b描述了VPIC强缩放和弱缩放的L3缓存丢失率和CPI。 虽然高速缓存丢失增加与强大的扩展OptaneDC执行，CPI仍然低于DRAM只执行。 内存模式执行的缓存丢失率比仅NVM和仅DRAM执行的速度要慢。 在弱缩放中，只有Optane的执行会导致L3缓存丢失的次数比只有DRAM和内存模式的执行少，但CPI较高。 这使得只有Optane执行的执行时间对于弱缩放保持在较低的水平。 正如在结果中所看到的，VPIC优化了它的缓存命中，以获得更高的性能，因此在所有三个执行中的执行时间都有很小的差异，。 这种应用可以受益于OptaneDC，减少能源消耗，同时不损害性能。

6.2.4 SNP结果

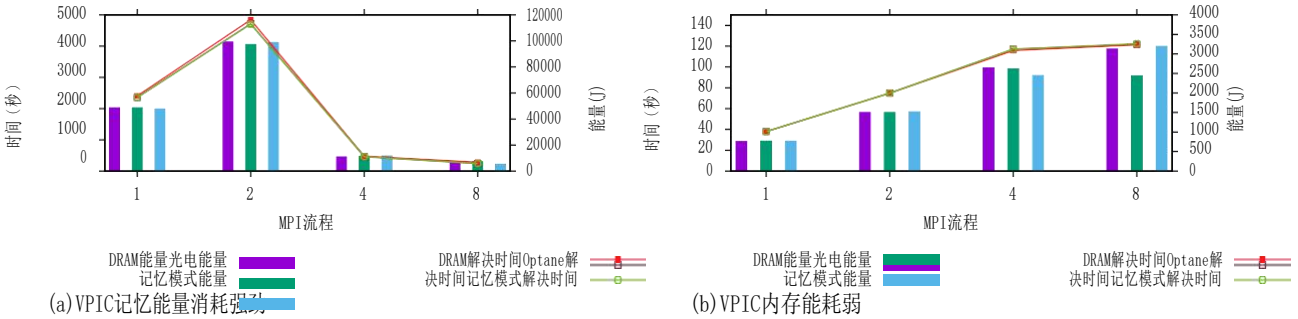
无花果。 17a还有图。 17b描述SNAP在光执行、DRAM执行和内存模式执行中的强和弱缩放的内存带宽和执行时间。 我们使用MPI将进程的数量从1扩展到8。我们使用用mpicc编译的SNAP的C版本。 在这里，我们观察到，在强缩放中的执行时间仅对所有三个执行略有变化。光电直流存储器带宽随着进程数量的增加而增加，在8个进程的所有三个执行中是最高的。 这反映在所有3次处决的执行时间中。 然而，对于弱缩放，当我们扩展到8个进程时，只有Optane执行的执行时间会增加。 我们还观察到，只有Optane执行的内存带宽不会随着弱缩放而增加。 这解释了与只有DRAM和内存模式执行相比，只有Optane执行的2倍高的执行时间。 无花果。 18a 还有图。 18b 描述SNAP执行在光执行、DRAM执行和内存模式执行中的能耗。 能量



(a) VPIC带宽，强尺度

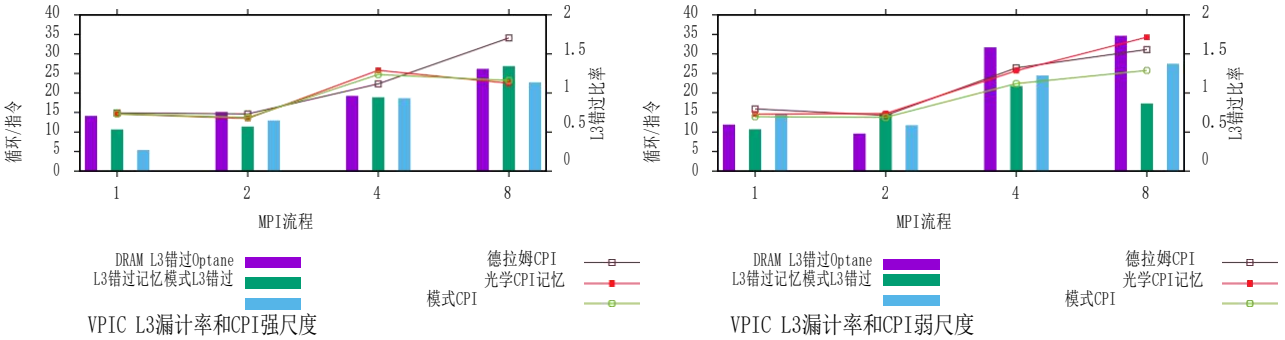
(b) VPIC带宽弱尺度

图14。VPIC的带宽测量



(a)VPIC记忆能量消耗弱孙 (b)VPIC内存能耗弱

图15。内存能耗为VPIC

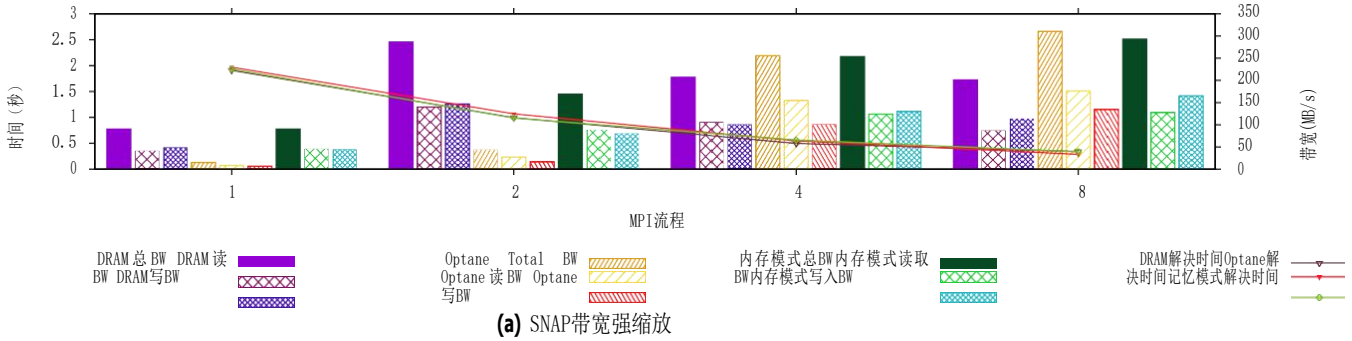


VPIC L3漏计率和CPI强尺度 VPIC L3漏计率和CPI弱尺度

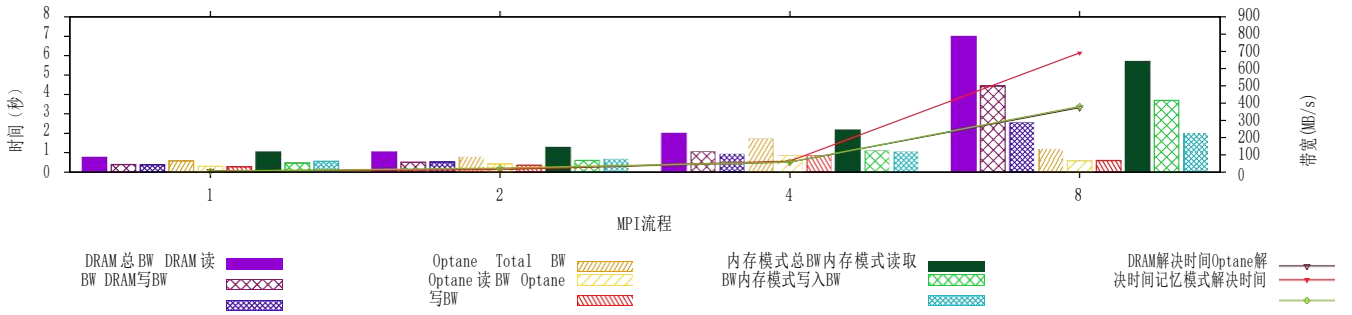
图16。VPIC的L3漏失率和CPI

消费在所有三个执行中保持相当恒定的强弱缩放。 由于这些执行的执行时间相似，只有Optane执行的能耗在3个执行中最少，当进程数为4时。

在弱缩放情况下，内存模式总体消耗的功率最小，但由于执行时间较长，它消耗的能量最大。 无花果。 19a还有图。 19b描述了SNAP强、弱缩放的L3缓存漏失比和CPI的图。 我们观察到CPI的差异显著

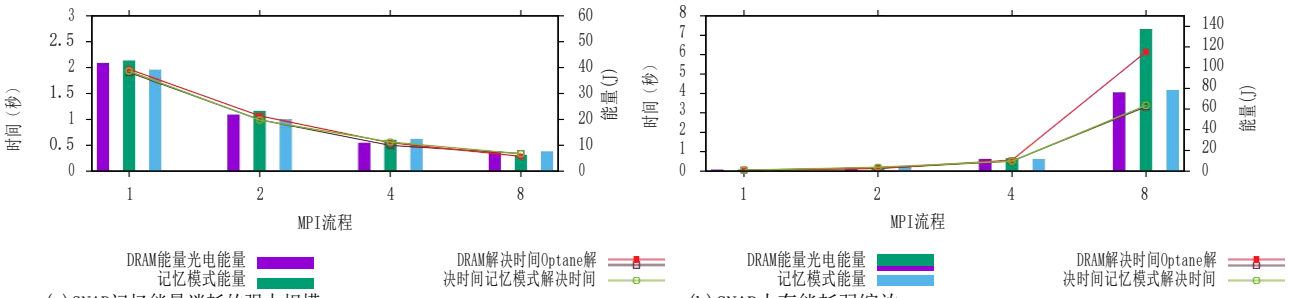


(a) SNAP带宽强缩放



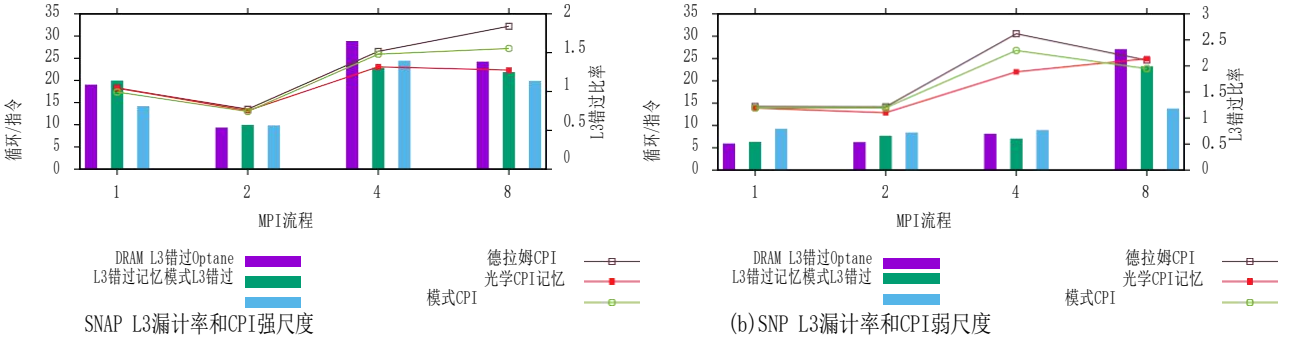
(b) SNAP带宽弱缩放

图17。用于SNAP的带宽测量



(a)SNAP记忆能量消耗的强大规模 (b)SNAP内存能耗弱缩放

图18。内存能耗为SNAP



SNAP L3漏计率和CPI强尺度 (b)SNP L3漏计率和CPI弱尺度

图19。SNAP的L3漏失率和CPI

在较高数量的进程中进行强缩放，其中只有Optane执行体验最少的CPI。 L3缓存丢失实际上随着所有三个执行的处理器数量的增加而增加。 L3缓存会随着处理数量的增加而忽略缩放，

也是。 然而，对于更高数量的过程，CPI的变化是不稳定的。 创建SNAP的C版本是为了利用Intel微架构中的向量操作，并对其进行了高度优化，以利用缓存层次结构和预取方法。因此，

使用SNAP，性能下降是最小的，这导致降低了低功耗光电直流存储器的能耗。

7 今后的工作

英特尔的OptaneDC PMM为在各种应用中使用NVM开辟了一系列可能性。 我们计划探索NVDIMMs在优化HPC应用中的应用。 您可以操作OptaneDCPMMS的不同模式有可能优化许多HPC应用程序。 根据我们的性能特性，我们计划为NVM开发不同配置的分配策略。 这将有助于解决计算节点较少的大问题，并在所需的计算和能源预算下运行。 Optane DC有许多可变组件需要描述，也需要充分利用该技术，例如其可变延迟和功耗。 我们计划研究NVDIMMs与HPC工作负载一起使用时的寿命和不同延迟及其对故障和故障的敏感性。 这将有助于提高使用NVDIMM的超级计算机的弹性。

除了大量的内存容量外，NVDIMM还具有数据持久性，这可以帮助开发新的弹性技术。 它们可以用来存储轻量级检查点和重新启动失败的进程。 我们计划探索为exascale超级计算机建立快速和轻量级检查点/Restart机制的可能性[22]. 它还可用于维护大型系统的元数据和帮助查找操作。 存储在NVDIMM上的数据可以通过使用校验和来提高可靠性来检测和纠正软错误。 我们将探索使用NVDIMM来提高计算的可靠性。 我们还计划研究有效使用NVDIMM所需的内核和用户级支持。 基于编译器的分析和分析信息可以帮助优化用于各种应用的NVDIMM。 我们还将评估对其他可以纳入DRAM-NVM混合内存层次结构的内存技术的支持。

8 结论

本文对由较慢的NVM设备和较快的DRAM设备组成的混合存储器系统进行了表征。 我们的结论是，由于较高的访问延迟和较低的内存带宽，使用较慢的字节寻址内存设备会阻碍内存绑定HPC应用程序的性能。 然而，使用DRAM作为速度较慢的NVM设备的缓存保持了仅在DRAM内存系统上观察到的HPC应用程序的性能，同时增加了系统的内存容量，这需要在大问题大小上进一步验证。 虽然使用NVM作为主存储器直接阻碍了性能的提高，具有合理权衡降低HPC应用能耗的潜力。 光学直流PMMS使我们能够缩小核心计数和内存容量缩放之间的差距。

致谢

这份材料是根据美国能源部国家核安全管理局主合同#89233218CNA000001于2018年11月1日为LosAlamos国家实验室分包合同#508854所支持的工作，NSF赠款1217748和1525609，以及美国合作的Exascale计算项目(17-SC-20-SC)所支持的工作。 能源部科学办公室和国家核安全局。

参考资料

[1] 德米特罗·阿帕尔科夫、阿列克谢·赫瓦尔科夫斯基、史蒂文·瓦茨、弗拉基米尔·尼基丁、唐学蒂、丹尼尔·洛蒂斯、吉索克·穆恩、小罗、陈尤金、阿德里安·奥恩格、亚历山大·德雷吉尔·史密斯和穆罕默德·克鲁尼。 2013. 自旋转移转矩磁随机存取存储器(STT-MRAM)。 *j. 紧急情况。 泰克诺。 Comput。 赛斯特。* 9、2、第13条（2013年5月），35页。 https://doi.org/10.1145/2463585.2463589

[2] Bowers、B J Albright、L Yin、W Daughton、V Roytershteyn、B Bergen和T J T Kwan。 2009. 用VPIC和Roadrunner模拟岩石动力学等离子体的研究进展。 *物理学杂志：会议系列180（2009年7月），012055。* https://doi.org/10.1088/1742-6596/180/1/012055

[3] Dhruva R Chakrabarti，Hans-J Boehm和Kumud Bhandari。 2014. Atlas：利用锁来实现非易失性内存的一致性。 在ACM SIGPLAN通知中，第一卷。 49. ACM，433-452。

[4] 乔尔·科伯恩、阿德里安·M·考菲尔德、阿米恩·阿克尔、劳拉·M·格鲁普、拉杰什·古普塔、兰吉特·贾哈拉和史蒂文·斯旺森。 2012. NV-Heaps：使用下一代非易失性存储器使持久对象快速和安全。 ACM计划通知47，4（2012），105-118。

[5] 莱昂纳多·达古姆和拉梅什·梅农。 1998. 打开MP：用于共享内存编程的行业标准API。 *IEEE Comput。 SCI。 英格。* 5，1（1998年1月），46-55。 https://doi.org/10.1109/99.660313

[6] Subramanya R Dulloor，Sanjay Kumar，Anil Keshavamurthy，Philip Lantz，Dheeraj Reddy，Rajesh Sankaran和Jeff Jackson。 2014. 用于持久内存的系统软件。 *第九届欧洲计算机系统会议记录。 ACM，15岁。*

[7] Gurbinder Gill、Roshan Dathathri、Loc Hoang、Ramesh Peri和Keshav Pingali。 2019. 使用英特尔光电直流持久存储器的大型数据集的单机图形分析。 共同RRabs/1904.07162（2019）。 阿希夫：1904.07162 http://arxiv.org/abs/1904.07162

[8] Saurabh Gupta、Tirthak Patel、Christian Engelmann和Devesh Tiwari。 2017. *大规模系统中的失败：长期测量、分析和影响。 国际高性能计算、联网、存储和分析会议记录。 ACM，44岁。*

[9] Hideto Hidaka、Yoshio Matsuda、Mikio Asakura和Kazuyasu Fujishima。 1990. 缓存DRAM体系结构：具有片上缓存内存的DRAM。 IEEE Micro10，2（1990），14-25。

[10] Joseph Izraelevitz，Jian Yang，Lu Zhang，Juno Kim，Xiao Liu，Amirsaman Memaripour，Yun Joon Soh，Ziuan Wang，Yi Xu，Subramanya R。 杜洛尔，赵吉申，史蒂文·斯旺森。 2019. 英特尔光电直流持久存储器模块的基本性能测量。 共同RRabs/1903.05714（2019）。 阿希夫：1903.05714 http://arxiv.org/abs/1903.05714

[11] 杰德克（2017）。 JEDEC DDR4SDRAM标准。 https://www.jedec.org/standards-documents/docs/jesd79-4a

[12] Sudarsun Kannan、Ada Gavrilovska、Karsten Schwan和Dejan Milojicic。 2013. 使用nvm作为虚拟内存优化检查点。 *在2013年IEEE第27届并行和分布式处理国际研讨会。 IEEE，29-40。*

[13] 伊恩·卡林，杰夫·基斯勒和罗布·尼利。 2013. *ESH2.0更新和更改。* 技术报告LLNL-TR-641973。 1-9页。

[14] 罗伯·拉瑟姆，N·米勒，罗伯特·罗斯，P·卡恩斯和克莱姆森·尤尼夫。 2004. Linux集群的下一代并行文件系统。 *Linux World Mag。* 2 (01 2004).

[15] 本杰明·C·李，EginIpek，OnurMutlu和DougBurger。 2009. 将相变内存构建为可伸缩的dram替代。 *ACM SIGARCH计算机架构新闻37，3（2009），2-13。*

[16] 徐丽，吕凯，王小平，徐周。 2012. NV-进程：基于非易失性存储器的容错进程模型。 亚太系统讲习班会议记录。 ACM，1。

[17] 凯文·林，张吉川，特雷弗·马吉，帕萨萨拉斯·兰甘纳坦，史蒂文·K·赖因哈特和托马斯·F·温尼斯奇。 2009.用于在刀片服务器中扩展和共享的分类内存。 *在ACMSI GARCH计算机体系结构新闻，第一卷。* 37. ACM，267-278。

[18] 约翰·D。麦克卡平。 1991-2007. *STREAM：高性能计算机的可持续内存带宽。 技术报告。* 弗吉尼亚大学，夏洛茨维尔，弗吉尼亚。 http://www. cs.virginia.edu/stream/ 不断更新的技术报告。 http://www. cs.virginia.edu/stream/。

[19] 约翰·D。 麦克卡平。 1995. 当前高性能计算机的内存带宽和机器平衡。 *IEEE计算机学会计算机架构技术委员会通讯(12月。*1995)，19–25.

[20] 奥努尔·穆特鲁。 2013. 内存缩放：系统架构透视图。 *在2013年第五届IEEE国际内存研讨会。* IEEE，21-25。

[21] 拉维·奈尔。 2015. 内存体系结构的演变。 *Proc。* IEEE103，8（2015），1331-1345。

[22] Onkar Patil 、 Saurabh Hukerikar 、 Frank Mueller 和 Christian Engelmann。 2017. 探索非易失性存储器的用例，以支持HPC的弹性。 *SC海报会议（2017年）。*

[23] Onkar Patil、Charles Johnson、Mesut Kuscu、Joseph Tuek、Tuan Tran和Harumi Kuno。 2009. 持续的区域，以生存NVM媒体失败。 (2009).

[24] Georgios Psaropoulos 、 Ismail Oukid 、 Thomas Legler 、 Norman May 和 Anastasia Ailamaki。 2019. 弥合NVM和DRAM之间的延迟间隙，用于延迟绑定操作。 *第15届新硬件数据管理国际研讨会论文集。* ACM。

[25] 西蒙娜·劳克斯，冯雄，马蒂亚斯·乌蒂格和埃里克·波普。 2014. 相变材料和相变记忆。 MRS公告39，8（2014），703a AS710。 https://doi.org/10.1557/mrs.2014.139

[26] 布莱恩·M·罗杰斯，阿尼尔·克里希纳，戈登·B·贝尔，肯·武，蒋小伟和严·索利欣。 2009. 缩放带宽墙：CMP缩放的挑战和途径。 *ACM SIGARCH计算机架构新闻37，3（2009），371-382。*

[27] 托马斯·舒尔、黄健和约瑟夫·托拉斯。 2019. 设计一个用户友好的JavaNVM框架。 （2019）。

[28] 2018年SICM。 *以记忆为中心的高级讲习班记录性能计算，MCHPC@SC2018，达拉斯，德克萨斯州，美国，2018年11月11日。 ACM。* http://dl.acm.org/citation.cfm?id=3286475

[29] SNP[n。 d.]。 SNAP：SN（离散坐标）应用代理。 https://github.com/lanl/SNAP

[30] 泰坦（2019年）。 泰坦。 https://www.olcf.ornl.gov/olcf-resources/ 计算系统/泰坦/

[31] TOP500名单-2018年11月（2018年）。 前500名名单-2018年11月。 https://www.top500.org/list/2018/11/

[32] 简·特雷比格，乔治·哈格和格哈德·韦尔林。 2010. likwid：x86多核的轻量级面向性能的工具套件环境。 *2010年第39届并行处理讲习班国际会议。 IEEE，207-216。*

[33] 亚历山大·范·雷宁，卢卡斯·沃格尔，维克托·莱斯，托马斯·诺依曼和阿方斯·珀。 2019. 持久内存I/O原语。 *AR XIV预印AR XIV：1904.01614（2019）。*

[34] 杰弗里·S·维特和斯帕什·米塔尔。 2015. 在极端规模的高性能计算中，非易失性存储器系统的机会。 *科学与工程中的计算17，2（2015），73-82。*

[35] Haris Volos，Andres Jaan Tack和Michael M Swift。 2011. 记忆：轻量级持久记忆。 在ACMSI GARCH计算机架构新闻，第一卷。 39.ACM，91-104。

[36] 王超，SudharshanSVazhkudai，小松马，费孟，金英杰和克里斯蒂安·恩格曼。 2012. NVMalloc：将聚合SSD存储作为极端规模机器中的内存分区。 *在2012年IEEE第26届国际并行和分布式处理研讨会。 IEEE，957-968。*

[37] 吴凯，弗兰克·奥伯，沙里·哈姆林，董力。 2017. 使用HPC I/O工作负载早期评估英特尔光机非挥发性存储器。 公司RRabs/1708.02199（2017）。 Ar Xiv：1708.02199 http://arxiv.org/abs/1708.02199

[38] 杨军，青松伟，程晨，王春东，海梁勇，和炳胜。 2015. NV-Tree：降低基于NVM的单级系统的一致性成本。 *第十三届USENIX文件和存储技术会议(FAST15)。* 167–181.

[39] Ulrike Meier Yang等人。 2002. Boomer AMG：并行代数多网格求解器和预条件。 *应用数值数学41，1（2002），155-177。*