

请参见本出版物的讨论、统计和作者简介，网址为：<https://www.researchgate.net/publication/280577113>

A Survey Of Techniques for Architecting DRAM Caches

2015 年 1 月 IEEE 并行和分布式系统事务中的文章

DOI: 10.1109/TPDS.2015.2461155

引文

34

读

1, 253

2 名作者:



Sparsh Mittal

Indian Institute of Technology
Roorkee

133 PUBLISHES 2, 050 引文



Jeffrey Vetter

University of Tennessee

258 PUBLISHES 7, 010 引文



本出版物的一些作者也在从事这些相关项目:



群论和量子学 [View project](#)



理解高级编程模型在当代异构架构上的可移植性 [View project](#)

动态随机存取存储器架构技术综述

IEEE 成员斯巴塞·米塔尔和 IEEE 高级成员杰弗里·维特尔

摘要——最近增加内核数量和内存/带宽墙的趋势导致了芯片架构的重大变革。面对日益增长的高速缓存容量需求，研究人员现在已经探索了动态随机存取存储器，它通常被认为是主存储器的同义词，用于设计大型末级高速缓存。然而，主流计算系统中的动态随机存取存储器高速缓存的有效集成也提出了一些挑战，并且已经提出了一些最新的技术来解决这些挑战。在这篇文章中，我们提出了一个架构动态随机存取存储器缓存技术的调查。此外，通过在几个维度上对这些技术进行分类，我们强调了它们的异同。我们相信，这篇论文将非常有助于研究人员深入了解动态随机存取存储器的潜力、权衡和挑战。

索引术语——回顾、分类、末级高速缓存、芯片堆叠、3D、堆叠动态随机存取存储器、带宽墙、超大规模系统、架构技术。



1 引言

随着正在进行的工艺技术扩展的探索遇到了功率和带宽墙的巨大挑战，现代处理器的架构已经见证了戏剧性的转变。随着芯片内核数量的增加，现代应用对高速缓存容量的需求不断增加，使用静态随机存取存储器设计高速缓存的传统智慧受到了严格的审查。静态随机存取存储器的低密度加上高泄漏功耗使得使用大型静态随机存取存储器高速缓存不可行，因此，现代处理器中最大静态随机存取存储器高速缓存的大小范围仅为几十兆字节（例如，Ivytown 处理器中的 37.5 兆字节静态随机存取存储器有限责任公司 [1]）。相比之下，现代横向扩展工作负载（如业务分析、网络搜索等。）需要数百兆字节的末级缓存 [2, 3]。为了避免带宽墙，还需要大型高速缓存，如果这种高速缓存使用静态随机存取存储器设计，它们可能会在未来的工艺技术生产中占据 90% 的芯片面积 [4]。这严重限制了可以放置在芯片上的内核数量，从而限制了多核扩展 [5]。此外，鉴于静态随机存取存储器的高泄漏功率，大型静态随机存取存储器高速缓存可能需要昂贵的冷却解决方案（例如，液体冷却），这显著增加了芯片的成本和复杂性。这些因素和趋势促使研究人员探索设计高速缓存的静态随机存取存储器的替代方案。

南米塔尔是未来技术集团，橡树岭国家实验室，橡树岭，TN 37830。电子邮件:mittals@ornl.gov.

J. 维特尔是未来技术集团，橡树岭国家实验室，橡树岭，TN 37830，和佐治亚州亚特兰大理工学院，GA 30332。电子邮件:vetter@ornl.gov.

这项工作得到了美国能源部科学办公室高级科学计算研究所的支持。

尽管与静态随机存取存储器相比，动态随机存取存储器提供了近 8 [6, 7] 的密度优势，但其相对较长的延迟通常限制了其仅用于设计主存储器。然而，最近在芯片堆叠方面的进步已经导致这些性能的显著改善 [6]。通过提高集成密度和消除布线，堆叠式动态随机存取存储器可以提供比片外动态随机存取存储器多 8 [8] 到 15 [9] 的带宽，同时只产生一半 [10, 11] 或三分之一 [12] 的延迟。

这些特性推动了几家领先厂商开发和发布高性能堆叠式动态随机存取存储器 [13–17]，因此，堆叠式动态随机存取存储器即将集成到中央处理器和图形处理器的存储器层次结构中 [9, 18, 19]。事实上，英特尔的 14 纳米骑士着陆处理器据说使用 16GB 堆叠动态随机存取存储器末级高速缓存，带宽为 500 GB/s [9]。虽然非易失性存储器，如 STTRAM（自旋转移矩随机存取存储器）和电阻随机存取存储器（电阻随机存取存储器）也被考虑用于设计末级高速缓存 [20]，但它们还没有达到制造成熟和商业可行性的水平，这可能证明它们的主流计算系统中的采用是合理的。显然，至少在可预见的未来，动态随机存取存储器将是一种有前途的技术，通过设计大型高速缓存来缓解功耗/带宽障碍。

然而，充分利用动态随机存取存储器的潜力需要解决几个挑战，并仔细选择设计参数，如块大小、标签存储架构、3D 分区策略、热管理问题等。事实上，由于静态随机存取存储器和动态随机存取存储器之间的显著差异，盲目地将甚至众所周知的静态随机存取存储器高速缓存优化（例如，最小化失败率，使用高关联性和更好的替换策略）移植到动态随机存取存储器

缓存可能会降低其性能[8]。显然，需要新的技术来充分实现动态随机存取存储器高速缓存的潜力。

在这篇文章中，我们综述了设计动态随机存取存储器的技术。我们根据关键参数对这些技术进行分类，以突出它们的异同。我们关注的是架构和系统级技术，而不是设备级技术。由于不同作品中使用的评估方法和工作量各不相同，我们只关注他们的关键研究见解，通常不讨论他们的定量结果。我们希望通过动态随机存取存储器缓存研究前沿的统一看法，为该领域未来的研究和发展提供明确的方向。本文中，DRAM 缓存一般指堆叠式 DRAM 缓存，因为片外(商品)DRAM 的较高延迟和较低带宽使其不适合缓存设计。

论文的其余部分组织如下。第 2 节简要介绍了动态随机存取存储器高速缓存及其设计中涉及的权衡。第 3 节首先介绍了动态随机存取存储器高速缓存管理技术的概述和分类。然后回顾了四种管理动态随机存取存储器的技术。最后，第三节

4. 本文最后讨论了未来的挑战。

2 背景

我们现在讨论使用动态随机存取存储器缓存背后的动机，以及在使用中遇到的挑战。关于动态随机存取存储器架构[21-24]和芯片堆叠技术[2, 6, 25]的更多细节，我们请读者参考以前的工作。

2.1 动态随机存取存储器的承诺

虽然动态随机存取存储器传统上被认为是与主存储器同步的，但最近的几个趋势提出了考虑将其用于设计高速缓存的令人信服的理由。

不断增加的内存压力和高密度的动态随机存取存储器：随着内核数量的增加，内存层次结构的压力也在增加。引脚数可扩展性差导致内存带宽墙，需要使用大容量缓存等有效方法来弥补。如前所述，静态随机存取存储器的低密度和高泄漏能量是其用于设计大型高速缓存的主要挑战。尽管架构技术可以部分抵消静态随机存取存储器[26]的能源挑战，但与极端规模存储系统的能效和容量要求相比，它们无法弥补数量级的差距。

从器件级特性来看，我们发现 SRAM 具有 6T (6 晶体管)结构，单元尺寸为 120-200F² (F 表示特征尺寸)，而 DRAM 具有 1T1C (一个晶体管和一个电容器)结构

细胞大小为 6-10F² [20, 23]。显然，动态随机存取存储器比静态随机存取存储器提供更高的密度。因此，动态随机存取存储器技术可以集成千兆字节大小的高速缓存，从而成为克服内存和带宽障碍的一种有前途的存储技术。可以利用动态随机存取存储器提供的密度优势在芯片上放置更多内核，这为多核扩展的继续铺平了道路[4, 5]。这些因素促使研究人员使用动态随机存取存储器，甚至将其用作 L4 高速缓存[27-31]和 NVM 主存储器的高速缓存[32]。

芯片堆叠的优势：如前所述，与传统的片外存储器相比，芯片堆叠提供了显著的延迟和带宽优势[6, 25, 33]。就每比特能量而言，基于 TSV (通过硅通孔)的接口比低功耗动态随机存取存储器接口 (LPDDR2) 提高了一个数量级，比 DDR3 接口提高了两个数量级[2]。通过在堆栈上共享外围电路，管芯堆叠的动态随机存取存储器也可以提高成本效率。此外，虽然堆叠多个处理器芯片可能会产生热量问题，但在处理器内核上堆叠内存只会导致温度小幅上升(低于 10 ° 为额外的 8 层内存[25])。

片外主存储器提供的带宽

由于引脚数量的限制而受到限制。芯片堆叠避免了引脚限制，因此提供了低延迟互连和非常高的带宽。同时，堆叠 DRAM 提供的容量明显小于片外 DRAM [3, 10, 34]，例如在英特尔的 Knights Landing 系统中，堆叠 DRAM 和片外 DRAM 的容量分别为 16GB 和 384 GB[9]。此外，堆叠动态随机存取存储器的带宽受堆叠动态随机存取存储器本身的并行性限制，而不受接口限制。这些事实使得堆叠式动态随机存取存储器更适合作为高速缓存，而不是主存储器。

维护软件透明性：与使用动态随机存取存储器作为主存储器相比，使用动态随机存取存储器作为高速缓存的好处是软件(应用程序和操作系统)不需要修改就可以获得高速缓存的好处。因此，遗留软件可以受益，并且不需要依赖操作系统供应商。

2.2 动态随机存取存储器设计中的因素和权衡

尽管有这些承诺，但动态随机存取存储器高速缓存也存在一些挑战，需要加以解决，以确保它们在产品系统中的有效集成。

性能挑战：对于局部性较差的应用程序，添加动态随机存取存储器缓存可能不会充分降低失败率。事实上，由于其延迟仅比主存储器低 50%，DRAM 缓存可能会显著增加未命中的延迟。容量之间也有权衡

(或未命中率)和等待时间,因此,一些技术旨在优化等待时间[8],而其他技术以牺牲其他参数为代价来优化未命中率[29]。动态随机存取存储器还需要刷新,这将消耗大量的能量并降低设备的可用性[24]。由于这些权衡,将动态随机存取存储器的大小增加到超过限制可能是不可行的。

架构和元数据管理:由于其大容量,DRAM缓存也需要大量的元数据,如标签、脏位、有效位等。(此后我们将“标签”和“元数据”作为同义词使用)标签查找位于所有请求的关键路径上。这使得选择高速缓存行粒度和标签的存储位置(例如在静态随机存取存储器或动态随机存取存储器中)成为动态随机存取存储器高速缓存的关键设计问题,它们也是相互关联的。两个极端是细粒度(也称为基于块的,粒度为64B)和粗粒度(也称为基于页的,粒度为几千字节)设计,这导致了截然不同的权衡。假设每条高速缓存线有6B的标记开销,对于细粒度(64B线)和粗粒度(2KB线)设计,1GB动态随机存取存储器高速缓存将分别需要96MB和3MB的标记存储。

很明显,细粒度的组织会产生非常高的标签存储开销,这可能远远大于末级静态随机存取存储器缓存本身的大小。这迫使设计者将标签存储在动态随机存取存储器本身中,这导致命中延迟增加。串行标签数据访问和集合关联高速缓存的使用进一步增加了这种延迟,因此,已经提出了并行标签数据访问和直接映射高速缓存的使用(见表1),尽管这些设计选择有其自身的局限性[26]。访问动态随机存取存储器以更新替换信息降低了动态随机存取存储器的可用性,因此,一些作品使用随机存取存储器替换策略而不是LRU(最近最少使用)策略,因为前者不需要更新[8]。一些技术使用智能预测器来绕过对动态随机存取存储器高速缓存的访问,这些请求被预测为未命中,它们的效率取决于预测的准确性和对脏数据的正确管理[35]。细粒度的组织在利用空间局部性方面也不太有效,因此会导致较高的失败率,从而导致片外访问的损失。细粒度组织的优势在于,它们可以有效利用片外带宽和缓存容量。

粗粒度的组织招致小得多的标签开销,这允许标签存储在静态随机存取存储器中,从而导致快速查找。它们依赖于利用空间局部性,因此,与细粒度缓存相比,它们可能会看到更高的命中率,尽管它们的失败率降低可能不足以减少带宽浪费。粗粒度方法的局限性在于,它会导致带宽浪费和队列争用,从而导致缓存层次结构中的背压延迟(仅写回脏子块会部分减少内存

交通[22])。大的高速缓存线导致高速缓存容量的利用不足,也导致多线程应用中的错误共享[29]。此外,粗粒度设计的高速缓存行粒度受到动态随机存取存储器页面大小的限制,因为使用更大的粒度将需要跨多个物理行映射数据,这将导致时序和能量效率的降低。此外,随着动态随机存取存储器缓存容量的增加,即使是粗粒度的组织也会产生越来越多的标签开销,这使得在静态随机存取存储器中存储标签变得不可行。

制造和可靠性挑战:由于动态随机存取存储器的独特设计,动态随机存取存储器高速缓存中的高速缓存控制器还需要执行设备级任务,例如管理刷新、发出行激活和预充电命令、满足动态随机存取存储器设备时间限制等。这些可能会导致显著的复杂性,并可能需要重新设计高速缓存控制逻辑[3]。此外,与传统的2D片外存储器相比,管芯堆叠存储器需要高得多的可靠性,因为管芯堆叠存储器中的故障可能不容易维修,因此可能需要丢弃包括功能处理器层的整个封装[36]。此外,设计具有非常大层数的存储器堆栈带来了功率输送和冷却、产量、可测试性等挑战。[25]。

混合高速缓存的挑战:一些研究人员提出了静态随机存取存储器/动态随机存取存储器混合高速缓存(参见表1)。由于静态随机存取存储器的延迟小于10ns[33, 37],而动态随机存取存储器的延迟为60-80ns[34, 37],静态随机存取存储器和动态随机存取存储器的延迟之间的巨大差异可能会使不同访问的命中延迟不同,这使得独立指令的调度变得困难。因此,在静态随机存取存储器/STTRAM或静态随机存取存储器/随机存取存储器混合高速缓存[38]中使用的数据迁移方案在静态随机存取存储器/动态随机存取存储器混合高速缓存中可能不可行。

3 动态随机存取存储器高速缓存管理技术

表1提供了本次调查中讨论的动态随机存取存储器高速缓存管理技术的概述和分类。该表根据研究技术的优化目标、关键特性和使用的缓存体系结构对其进行了分类。这种分类有望对计算机辅助设计人员、研究人员和技术营销专业人员有用。

我们现在简要讨论几种动态随机存取存储器高速缓存管理技术。为了方便起见,我们将它们大致分为几类。

3.1 芯片堆叠动态随机存取存储器的比较评估

在这一小节中,我们将讨论探索动态随机存取存储器缓存体系结构和/或将其与静态随机存取存储器缓存进行比较的研究工作。

表1
研究工作的分类

分类	参考
研究/优化目标	
表演	[3, 4, 6-8, 10-12, 21, 27-32, 35, 39-58]
能源节省	[2, 6, 7, 32, 43, 47, 49, 53, 54, 56, 57, 59-61]
提供内核间 isola- 象征式互动	[42, 44, 45]
集成节省面积 更多内核	[2, 4, 5, 62]
刷新开销管理- 名词后缀, 表示“...的行 为”	[59, 61]
动态优化 带宽使用	[8, 12, 21, 32, 34, 35, 53, 58]
热量管理	[2, 6, 60]
可靠性/弹性	[36, 60]
减轻过程变化	[50]
与静态随机存取存储器的比较 隐藏所	[4, 6, 7, 32, 42, 47, 56, 57, 59, 62]
与 NVM 的比较 隐藏所	[56, 57]
缓存体系结构	
静态随机存储器+动态随机存 储器混合高速缓存	[43, 44, 49]
直接映射缓存	[3, 8, 32, 36, 48, 58]
并行标签数据访问	[8, 21, 36, 51, 58]
动态随机存取存储器上的元数 据	[3, 8, 10, 11, 21, 29, 31, 32, 36, 39, 41, 44, 46, 48, 50-52, 54, 55, 58]
静态随机存取存储器元数据	[6, 12, 39, 43, 49, 53, 61]
细颗粒	[3, 8, 10, 29, 31, 36, 39, 41, 44, 48, 49, 52, 54, 58]
粗(比块大)- 小麦	[6, 12, 28, 50, 51, 53, 55]
自适应粒度	[21, 32]
技术的关键特征	
控制对动态随机存取存储器的 访问	[22, 35]
高速缓存或主存储器 作为高速缓存或 操作系统管理的主内存	[3, 10, 30, 48]
缓存重新配置	[43, 49, 54]
缓存映射方案	[44-46]
使用额外的结构/方法	
命中/未命中或位置预测- tor 或标签缓存	[8, 10, 21, 21, 27, 31, 32, 35, 39, 39]

Black 等人[6]从性能和散热角度评估了不同的 SRAM 和 DRAM 2D/3D 高速缓存设计。他们的基准是带有 4MB 静态随机存取存储器高速缓存的双核处理器。对于第二种设计, 假设 8MB 堆叠 L2 的面积与基线芯片的面积几乎相同, 他们将 8MB 静态随机存取存储器缓存直接堆叠在处理器芯片上方。假设动态随机存取存储器密度是静态随机存取存储器密度的 8 倍, 在第三种设计中, 4MB 静态随机存取存储器 L2 被替换为 32MB 堆叠动态随机存取存储器 L2, 并且节省在处理器管芯上的面积用于存储动态随机存取存储器高速缓存的标签。在第四种设计中, 64MB 的动态随机存取存储器缓存堆叠在顶部, 其标签存储在处理器芯片上的 4MB L2 中。他们表明, 由芯片堆叠动态随机存取存储器高速缓存实现的高速缓存容量的增加有助于提高性能和能效, 并减少片外带宽。他们还指出, 与

基线, 没有一个堆叠的高速缓存设计会显著影响热分布, 尽管与 DRAM 相比, 由于 SRAM 的功率密度更高, 堆叠 SRAM 会导致更高的热增加。

Sun 等人[41]提出了一种多核系统的设计, 其中 3D DRAM 被用于为每个核实现私有的 L2 高速缓存以及由所有核共享的主存储器。为了设计 3D 动态随机存取存储器而不要求对 TSV 制造有严格的限制, 他们提出了一种粗粒度的 3D 分区策略。他们建议保留单个存储器子阵列列为单层, 而不是将单个存储器子阵列分成几层。他们表明, 随着动态随机存取存储器芯片数量和 L2 高速缓存容量的增加, 3D 动态随机存取存储器 L2 高速缓存的延迟可能与 2D 静态随机存取存储器 L2 高速缓存相当, 甚至更小。为了进一步降低动态随机存取存储器的延迟, 他们建议减小每个单独的动态随机存取存储器子阵列的尺寸, 并在外围电路和 H 树缓冲器中使用低阈值电压晶体管。他们表明, 与使用 3D 动态随机存取存储器作为主存储器的基线相比, 他们将 3D 动态随机存取存储器用于高速缓存和主存储器的方法显著提高了性能。

徐等[7]研究了不同片上网络 (NoC) 设计的 SRAM 和 DRAM LLCs, 从访问延迟、缓存大小、面积和功耗等方面评估 DRAM 缓存的效果。与 2D 静态随机存取存储器有限公司相比, 2D 动态随机存取存储器有限公司和三维动态随机存取存储器有限公司降低失败率, 这也减少了路由器/链路活动的总量。然而, 与 2D 动态随机存取存储器有限责任公司相比, 3D 动态随机存取存储器有限责任公司由于更短的导线长度而显示出更低的延迟, 但是由于路由器和链路数量的增加以及路由器复杂性的增加而显示出更高的功耗。

3.2 管理元数据的技术

赵等[39]评估了多个动态随机存取存储器高速缓存设计延迟/面积/带宽的权衡, 即。1) 使用动态随机存取存储器中的标签, 2) 使用静态随机存取存储器中的标签, 3) 在静态随机存取存储器中具有标签的分区高速缓存[63], 4) 在静态随机存取存储器中使用“部分标签”, 5) 分区高速缓存和部分标签的组合。对于标记在静态随机存取存储器中的方法, 标记存储在最后一级静态随机存取存储器高速缓存的一部分中, 这降低了其数据存储容量。扇区化动态随机存取存储器高速缓存使用大粒度块, 由多个子块和块地址组成。每个子块由小缓存块及其状态组成。分区的动态随机存取存储器高速缓存通过使用大粒度块来减少标签开销, 并且通过在每次未命中事件中仅带来子块而不是整个扇区来减少高速缓存流量。为了抵消其高失败率, 还使用了预取。部分标签方案在静态随机存取存储器中保持很少的标签最低有效位(即部分标签), 而在动态随机存取存储器中保持完整的标签目录。通过查阅部分标签, 可以进行命中/未命中的预测, 以避免在预测的未命中时访问动态随机存取存储器。他们展示了将分区高速缓存与部分

标签提供了最高的性能提升，同时还减少了标签开销。

Jiang 等人[12]提出了基于滤波器的技术来避免基于页的 DRAM 高速缓存中的带宽浪费。他们的第一种技术使用过滤式高速缓存来分析页面，并选择在动态随机存取存储器高速缓存中分配的热点（即最常访问的）页面。通过不在动态随机存取存储器高速缓存中分配低空间局部性的页面，他们的技术避免了带宽浪费。这种技术的局限性在于，当受害页面从过滤器缓存中被逐出时，其访问历史会丢失，并且在以后的访问中，该页面被认为是新的，因此，许多页面在达到被认为是热的阈值之前被逐出。为了避免这种情况，第二种技术将从筛选器缓存中逐出的页面的计数器存储到内存中，并在以后访问该页面时恢复它。因此，第二种技术提高了热页识别的准确性，这允许减小过滤器高速缓存的大小。他们的第三种技术通过基于应用或其阶段是否（分别）使存储器带宽饱和来动态激活或去激活过滤器高速缓存，从而适应应用内和应用间的变化。当过滤器缓存被停用时，所有页面都被分配到动态随机存取存储器缓存中。

Loh 等人[29]提出了一种能够在动态随机存取存储器高速缓存中使用传统块大小（64B）的技术。它们将标记存储在动态随机存取存储器中，并将标记和数据访问安排为复合访问，以确保数据访问总是导致行缓冲区命中。为此，可以存储多达 32 个 64B 数据块的 2KB 动态随机存取存储器行被重组，使得它存储 29 个 64B 数据块以及这些数据块中每一个的 6B 标签（18B 未被使用）。因此，以动态随机存取存储器容量为代价，他们的技术通过将标签存储在动态随机存取存储器中来提供可扩展的标签管理方法。在动态随机存取存储器高速缓存未命中时，会产生动态随机存取存储器访问的开销，为了避免这种情况，他们维护一个称为 MissMap 的结构，该结构跟踪动态随机存取存储器高速缓存的内容。当 MissMap 确定缓存未命中时，将跳过对缓存的访问。他们的方法的局限性是 MissMap 的大小很大，因此将其存储在最后一级静态随机存取存储器缓存中是没有吸引力的。请注意，其他一些作品也将 MissMap 用于 DRAM 缓存管理[11, 44 - 46]。

Sim 等人[35]指出，MissMap 设计中使用的维护关于 DRAM 缓存内容的精确信息的关键思想过于保守。为了避免错误映射的开销，他们使用低成本的命中/未命中预测器，这有助于推测请求是否可以由动态随机存取存储器高速缓存或主存储器服务。他们还使用负载均衡方案，根据两个存储器的瞬时排队延迟，将存储器请求重定向到芯片堆叠的动态随机存取存储器高速缓存或芯片外主存储器。这有助于通过利用空闲的片外资源来提高整体系统带宽

动态随机存取存储器高速缓存服务于高速缓存命中突发时的带宽。这些技术的局限性在于，在存在脏数据的情况下，它们可能会导致不正确的执行，因为它们可能会访问主内存中的陈旧值。为了解决这个问题，他们使用混合写策略，在直写模式下操作大部分动态随机存取存储器缓存，并且只允许对少数热页进行回写。这确保了动态随机存取存储器高速缓存大部分是干净的，并且只对脏页的请求需要进入动态随机存取存储器高速缓存。这消除了等待预测器验证的需要，并允许更多机会将请求重定向到片外存储器。当 MissMap 的延迟和存储开销不可接受，工作负载是写无关紧要的，但受益于其负载平衡方法实现的更大系统带宽时，他们的技术非常有用。

为了避免 MissMap 的开销，El-Nacouzi 等人[31]提出了一个预测器来估计 DRAM 缓存中的命中/未命中。他们指出，由于空间局部性，首先访问一个页面，然后访问同一页面中的其他块。此外，在任何时候，只有很少的页面会经历未命中，并且这些未命中中很可能在第一次未命中后不久发生。因此，跟踪少量页面及其块可能会提供合理的预测精度。基于此，他们的技术使用了两个过滤器，即。粗颗粒和细颗粒。粗粒度过滤器跟踪在动态随机存取存储器高速缓存中具有高速缓存块的页面的超集，而细粒度过滤器跟踪选定页面中的所有块。他们的技术首先检查粗粒度过滤器，如果预测页面将被缓存，则检查细粒度过滤器以查看是否预测该块将被缓存。如果任何过滤器预测到未命中，将绕过对动态随机存取存储器高速缓存的访问。

Qureshi 等人[8]注意到，由于 DRAM 高速缓存比传统高速缓存慢，降低已经很高的命中延迟的优化（例如，使用 MissMap [29] 和标签数据串行化）可能会降低性能，即使它们在命中率方面提供了小的提高。因此，动态随机存取存储器缓存应该首先针对延迟进行优化，然后针对命中率进行优化。他们建议使用直接映射的动态随机存取存储器缓存来代替集合关联缓存，这也提高了行缓冲区命中，从而进一步降低了访问延迟。此外，为了避免标签序列化延迟，每次访问时都会一起检索数据和标签。他们还使用内存访问预测器来估计数据是否存在于动态随机存取存储器缓存中。如果是这样，则在访问主存储器之前访问动态随机存取存储器高速缓存，以节省带宽。如果数据不太可能出现在动态随机存取存储器高速缓存中，则动态随机存取存储器高速缓存和主存储器都被并行访问，以避免高速缓存未命中损失来减少延迟。

为了将基于块和基于页的高速缓存设计的优点结合在一起，Jevdjic 等人[53]提出了一种 DRAM 高速缓存设计，它以页的粒度分配数据，但只获取那些块

在页面驻留在缓存期间将被触及的页面内。他们的技术还识别不显示空间或时间局部性的页面，并且不在缓存中分配这些页面。为此，他们的技术使用了一个预测器，该预测器根据数据和访问这些数据的代码之间的高度相关性来识别页面的足迹。基于此，访问页面的第一条指令提供关于页面包含的数据的提示，并且通过监视和记住代码进一步访问哪些块，可以预测当同一段代码访问另一个页面时将需要哪些块。他们表明，他们的技术实现了高命中率、小查找延迟和低标签阵列开销，同时还消除了基于页面的设计的片外流量开销。

Jevdjic 等人[51]旨在将两种先前的方法[8, 53]的优点结合起来。它们使用基于页面的缓存分配来实现高命中率和低标记开销，同时仅估计和获取每个页面中的有用块，以最大限度地减少带宽浪费。它们使用集合关联缓存和路预测器，避免并行获取所有路。这避免了直接映射缓存的需要，直接映射缓存会导致冲突未命中。此外，它们保持每页单个标签，以简化足迹跟踪，并且标签读取与数据读取重叠，以避免标签序列化延迟。

Hameed 等人[46]提出了一种动态随机存取存储器高速缓存架构，其中每个动态随机存取存储器行包括四个高速缓存组，由1个标签块和7个数据块组成(即7路组关联性)。一行中的动态随机存取存储器高速缓存集数由存储块地址的两个最低有效位决定。由于查找的标签数量少于未命中率优化设计，并且具有更高的行缓冲区命中率，他们的技术实现了更低的动态随机存取存储器命中延迟。此外，与直接映射的动态随机存取存储器高速缓存相比，它通过使用7路关联高速缓存提供了更低的未命中率。由于对不同高速缓存集的访问的非均匀分布可能会增加冲突未命中，他们提出了一种集映射策略，以循环方式将动态随机存取存储器的行号分配给每个内核。这导致对不同组的动态随机存取存储器高速缓存的访问的均匀分布，这降低了未命中率。

3.3 使用标记缓存

Huang 等人[52]注意到，虽然静态随机存取存储器中的标签方法比动态随机存取存储器中的标签方法产生更高的面积开销，但是其性能是优越的。为了将两者结合在一起，他们的技术将标签保存在动态随机存取存储器中，但也将一小部分标签缓存在专用的静态随机存取存储器缓存中，称为聚合标签缓存。ATCache 通过缓存最近访问的标签来利用节奏局部性

仅通过预取相邻动态随机存取存储器高速缓存集合的标签，仅当集合之间的这种局部性已经被确认时，才预取动态随机存取存储器集合和空间局部性。他们表明，由于ATCache的访问延迟小，他们的技术比动态随机存取存储器中的标签方法获得了更好的性能。在所有的标记在静态随机存取存储器中的方法中，高速缓存仅产生一小部分面积开销，但是具有复制(由于高速缓存)静态随机存取存储器中的一些标记的缺点，这些标记也在动态随机存取存储器高速缓存中，并且这种开销随着动态随机存取存储器高速缓存大小的增加而增加。因此，当芯片空间的最终浪费和标签管理的更高复杂性对于实现与具有所有静态随机存取存储器中标签的性能相当的性能是可接受的时，他们的技术和缓存芯片上标签的类似技术(例如[27, 32])是有用的。

Hameed 等人[27]提出了一种解决大型动态随机存取存储器高速缓存的高标签延迟的技术。对于具有大的L3静态随机存取存储器高速缓存和L4动态随机存取存储器高速缓存的系统，他们提出了称为静态随机存取存储器标签高速缓存和动态随机存取存储器标签高速缓存的小静态随机存取存储器结构，它们分别保存最近在L3和L4中访问的集合的标签。这些标签缓存可以快速识别大型缓存的命中/未命中，从而减少标签查找延迟。对于空间局部性有限的应用程序，标记缓存中的命中率仍然很小。为了避免这种情况，他们的技术根据最近的行为来识别标签缓存中无用插入的数量，并限制这种插入以提高命中率。

3.4 自适应数据粒度的使用

Meza 等人[32]将元数据存储于动态随机存取存储器中与其数据相同的行中，以避免动态随机存取存储器缓存命中时的行缓冲区未命中。为了进一步减少元数据查找延迟，他们将最近访问的动态随机存取存储器行的元数据缓存在一个小的片上缓冲区(缓存)中。由于数据访问的局部性，热数据的元数据很可能被缓存在芯片上，因此，它们可以以与静态随机存取存储器标签存储相同的延迟被访问。与“满”静态随机存取存储器标签存储相比，它们的“热”静态随机存取存储器标签缓冲器的存储开销小三个数量级。为了实现局部性和带宽消耗之间的平衡，他们还建议动态调整迁移粒度。通过测试不同的粒度，例如128B、256B、512B、1KB、2KB、4KB和无迁移，他们的技术决定了线程的正确迁移粒度。这提高了动态随机存取存储器的高速缓存利用率，减少了带宽争用，但代价是自适应迁移粒度的复杂性增加。

Gulur 等人[21]提出了一种以双模方式组织数据的技术，使得具有高空间局部性的块被组织为大块，而那些具有有限空间局部性的块被组织为小块。通过在运行时为不同的块动态选择合适的存储粒度，它们的

该技术有效地利用了动态随机存取存储器的高速缓存容量，并减少了片外存储器的带宽消耗。每套可以装 X 大 (512B) 和 Y 小 (64B) 块。表示集合的状态为 (X, Y)，一个 2KB 的集合可以有状态 (4, 0)，(3, 8)，(2, 16)。因此，集合关联性 (=X+Y) 可以在每个集合的基础上动态变化，但是集合的数量是固定的。通过跟踪采样集中 64B 子块的利用率，他们的技术决定了采样方式应该被分类为大块还是小块。与 Loh 等人[29]的方案不同，在该方案中，元数据与同一动态随机存取存储器行上的数据交错，它们将属于一个通道的数据组的元数据存储到另一个通道的数据组上，这使得每页能够打包更多的元数据项，从而提高行缓冲区命中率。同时，由于堆叠动态随机存取存储器的高内部带宽，它允许元数据和数据的并发访问。为了进一步减少缓存命中延迟，他们的技术使用了一个小型的基于静态随机存取存储器的路定位器，它缓存了对动态随机存取存储器缓存集的最近访问的路标识。在命中路线定位器上，避免了元数据对动态随机存取存储器的访问。他们表明，他们的技术比动态随机存取存储器中的标签和静态随机存取存储器中的标签方法都提高了平均访问延迟。他们技术的局限性来自于以双模式方式组织数据的额外复杂性。

带宽节省方法的比较:值得注意的是，Jiang 等人[12]在 DRAM 中使用自适应页面分配来优化带宽，这不同于 Meza 等人[32]和 Gulur 等人[21]使用自适应获取粒度来优化带宽。Sim 等人[35]和 Gulur 等人[22] (参考第 3.13 节)除了利用芯片堆叠的 DRAM 高速缓存的带宽之外，还通过利用空闲的片外带宽来优化带宽。Qureshi 等人[8]通过一起读取标签和数据(即避免标签数据串行化)并使用直接映射高速缓存来优化带宽，该高速缓存避免了像在集合关联高速缓存中那样传输多个标签。Jevdjic 等人[53]使用基于页面的高速缓存设计，但仍然通过从页面中提取选定的块来节省带宽，当页面驻留在高速缓存中时，这些块将被访问。

3.5 探索二级记忆

Chou 等人[10]提出了一种方法，将堆叠动态随机存取存储器既用作主存储器的一部分以增加其容量，又用作高速缓存以捕获数据局部性并以细粒度执行数据管理。这种方法对于大尺寸的堆叠内存尤其有用，例如 4GB 堆叠内存和 12GB 主内存。他们的技术将最近访问的数据线存储在堆叠的动态随机存取存储器中，并将受害线交换到主存储器。因为这会动态地改变线路的物理位置，所以他们也使用线路位置表(LLT)

跟踪所有数据线的物理位置。LLT 类似于传统高速缓存的标签目录，除了 LLT 还识别主存储器中的行的位置，以防在堆叠的动态随机存取存储器中找不到它。LLT 与堆叠动态随机存取存储器中的数据位于同一位置，以避免静态随机存取存储器的存储开销。对于未存储在堆叠动态随机存取存储器中的行，LLT 查找和存储器访问被串行化，为了避免这种情况，它们使用行位置预测器。如果该预测器估计该行在主存储器中，则它与 LLT 访问并行地从主存储器中被访问。他们表明，他们的技术比使用堆叠动态随机存取存储器作为高速缓存或主存储器的一部分提供了更大的改进。

Meswani 等人[48]指出，考虑到芯片堆叠 DRAM 提供的有限能力，仍然需要片外存储器来满足大规模应用的存储器需求。这导致了两级存储器(TLM)设计，其要求存储器架构和软件应用的共同设计。为了管理这样的 TLM，他们研究基于硬件、基于操作系统和程序员驱动的方法，并比较它们的相对优势。硬件方案采用堆叠式动态随机存取存储器作为末级高速缓存。基于操作系统的方法将此用作主存储器，片外存储器用作传统的交换设备。在程序员驱动的方法中，关于多个存储器存在的信息被展示给程序员，程序员可以决定在不同的存储器上分配或固定不同的数据对象。他们使用 exascale 代理应用程序的研究表明，硬件缓存实现了高命中率，并且消耗了小带宽。程序员管理的 TLM 也消耗很少的带宽，但是命中率很低，而基于操作系统的方法由于使用了页面粒度分配而浪费了带宽。

3.6 解决与多核处理器相关的问题

Hameed 等人[45]指出，大型 DRAM 缓存中的应用程序不知道存储体映射策略会导致某些应用程序不公平的减速。为此，他们的技术跟踪并发执行的应用程序的运行时失败率信息，以检测抖动应用程序。此外，高速缓存存储体被组织成两个区域，即“私有区域”和“共享区域”。由于抖动应用程序可能会驱逐其他应用程序的有用高速缓存行，因此它们的技术旨在将抖动应用程序的内存段限制在单个高速缓存存储体中。这减少了内核间缓存的占用，并在抖动和非抖动应用程序之间提供了性能隔离。此外，为了提高高速缓存的利用率，共享区域被分配给提供协作高速缓存共享的无抖动应用。

Loh [42]提出了一种动态随机存取存储器高速缓存组织方案，以提高其性能。每套

被组织为多级逻辑队列，其中单个高速缓存行构成队列条目。所有高速缓存行最初都被插入到一级队列中。在向大小为 Q ($Q < W$ ，其中 W 是集合关联性) 的队列插入 Q 之后，原始行离开该队列。如果它在驻留在该队列期间没有收到任何命中，它将被逐出，否则，它将被插入下一(第二)级队列(依此类推)。最后一个队列使用 LRU 替换策略。使用这种组织，看到没有重用的行仅在 Q 插入后被驱逐，而在传统的 LRU 替换策略中，它将在 W 插入后被驱逐。这通过确保快速驱逐死线来提高高速缓存的效率。看到突发访问的行在驻留在一级队列中时满足这些请求，在突发结束后，这些行从二级队列中被逐出。因此，使用多级队列背后的关键思想是，它们充当有限重用的缓存访问模式的过滤器。在多核处理器中，每个内核都有自己专用的一级队列，二级队列可以共享。通过将来自不同内核的流量隔离到不同的一级队列中，他们的组织还为这些内核提供了性能隔离，因为具有流行行为的内核无法替换其他内核的线路。替换逻辑和队列设计的复杂性增加是这种技术的局限性。

Hameed 等人[45]和 Loh [42]的技术之间的区别在于，为了提供内核间隔离，前者使用了一种存储体映射策略，该策略明确限制了抖动应用程序的缓存配额，而后者使用了一种缓存替换策略，该策略间接加速了从抖动应用程序或很少重用的应用程序中驱逐行。因此，前者使用“严格分配”，而后者使用“软分配”。

3.7 支持多核扩展的动态随机存取存储器缓存

Rogers 等人[4]研究了由于带宽墙问题如何限制多核扩展，以及带宽节约技术在多大程度上可以缓解这一挑战。他们评估了几种技术，例如使用较小尺寸的内核来实现较大尺寸的缓存、3D 堆叠缓存、缓存压缩和链接压缩以及动态随机存取存储器缓存。

在比例缩放下，芯片上的内核数量应该在四代技术中达到 16 ($= 24$)，然而，带宽墙将其限制为只有 11 个内核(假设静态随机存取存储器 L2 缓存)。他们表明，假设动态随机存取存储器的密度是静态随机存取存储器的 4 倍和 8 倍，可支持的内核数量

分别达到 16 岁和 18 岁。尽管他们的模型做了一些简化假设(例如忽略刷新开销等。)，他们的结果证实了使用 DRAM 缓存是过滤大量内核产生的额外内存流量的有效方法

用于支持多核扩展。他们还表明，通过结合使用动态随机存取存储器缓存和其他带宽节约方法(如上所述)，带宽墙问题可以延迟几代技术，从而允许多核扩展继续。

Hardavellas 等人[5]指出，由于电源墙和片外带宽墙，内核数量的增加并不能转化为相应的性能提升。他们表明，通过使用定制的高能效异构多处理器可以部分克服功率墙，为了克服带宽墙，可以使用大型 3D 堆叠动态随机存取存储器缓存。带宽墙的限制提供了扩展空间，允许在芯片上增加更多的内核来进行性能扩展，尽管在这种情况下，网络子系统能耗的增加成为新的瓶颈。换句话说，使用动态随机存取存储器缓存的好处是有限的。

Pan 等人[62]研究了在 VLIW(超大规模指令字)处理器中使用管芯堆叠的 DRAM 高速缓存。他们指出，芯片堆叠使 3D 动态随机存取存储器缓存的延迟接近 2D 静态随机存取存储器缓存的延迟。利用这一点，他们建议将 2D 静态随机存取存储器缓存替换为 3D 动态随机存取存储器缓存，并将节省的区域分配给其他集群，以提高并行性和性能。它们表明，具有 3D 动态随机存取存储器 L2 高速缓存和 3D 动态随机存取存储器主存储器的四集群系统占据与具有 2D 静态随机存取存储器 L2 高速缓存和 3D 动态随机存取存储器主存储器的两集群系统相似的逻辑管芯面积，并且前者也提供更好的性能。

因此，尽管 Rogers 等人[4]、Hardavellas 等人[5]和 Pan 等人[62]使用不同的模型或评估平台，但他们都提供了相同的结论，即 DRAM 缓存可以有效地允许多核扩展继续进行。

3.8 更新管理技术

Jaksic 等人[59]提出了一种基于 DRAM 的 L1 和 L2 相干高速缓存设计。为了减少刷新开销，基于块的相干状态来决定块的刷新操作。例如，对于 MESI(修改的、排他的、共享的、无效的)一致性协议，它们表明无效行不需要刷新，处于共享或排他状态的行如果不刷新但不需要写回，则转换到无效，而处于修改状态的行如果不刷新则需要无效并写回。基于可刷新或不可刷新的行状态的组合，他们提出了多种刷新策略，从不刷新行到刷新修改的、共享的和独占的行。通过选择合适的刷新策略，可以实现性能损失和节能之间的折衷。他们还表明，与基于静态随机存取存储器的高速缓存设计相比，基于动态随机存取存储器的高速缓存设计

提供显著的节能效果，但性能损失很小。

戈什等人[61]提出了一种降低 3D 动态随机存取存储器高速缓存刷新要求的技术。他们注意到，从数据保持的角度来看，对存储器行的访问执行等同于定期刷新的操作，因此，定期刷新所有存储器行以保持数据的传统刷新机制导致能量的浪费。基于此，他们的技术为存储器模块中的每一行维护计数器，并且如果存储器行最近经历了读/写访问，则避免刷新该存储器行。

Jaksic 等人的技术之间的差异。

[59]和 Ghosh 等人[61]的观点是，前者使用高速缓存一致性的属性来跳过刷新操作，而后者从刷新的角度来看，使用访问操作到刷新操作的等价性来跳过对刚刚看到访问操作的行的刷新操作。

3.9 热管理技术

Milojevic 等人[2]研究了面向数据中心市场的片上服务器架构。芯片底层有众核计算引擎，顶层有宽 I/O DRAMs(类似于[13])，作为 LLC 供所有内核共享。为了优化总芯片吞吐量，底层为二级缓存分配了最少的面积，这刚好足以捕获数据密集型商业工作负载的指令和数据工作集。更大一部分面积专用于许多内核。他们使用以中央处理器为中心的工作负载(例如 SPECInt 和 Dhrystone)进行的评估表明，片上服务器(逻辑+动态随机存取存储器)的温度范围为 175–200°C，超过了可靠的工作范围。相比之下，云工作负载因其受内存限制的特性而在处理内核中消耗更少的功率，因此，使用云工作负载的评估显示，热点和温度在堆叠动态随机存取存储器的工作范围内时，功率密度会降低。因此，他们的研究表明，即使采用低成本冷却方案，片上服务器系统也是可行的，这为数据中心节约成本和能源提供了机会。

Yun 等人[60]指出，3D 芯片的高集成密度可能导致高工作温度，这增加了泄漏功耗和错误率。为此，他们提出了一种 3D 堆叠动态随机存取存储器高速缓存的 DVFS 方案，其中每个高速缓存存储体或每组高速缓存存储体的电压/频率可以基于错误率(由于保留和感测失败)、高速缓存访问率和温度引起的功耗来调整。他们的技术在运行时监控每个时间间隔内的缓存访问速率和缓存区温度。对于交流频率较低的应用，使用低电压和低频率来降低

功耗低，性能损失小，反之亦然。因为差错率随着电源电压的降低而增加，所以电压的最小值也由差错率约束决定。

因此，Milojevic 等人[2]研究并比较了以 CPU 为中心和云工作负载的温度分布，而 Yun 等人[60]提出了一种基于的热管理技术。

3.10 确保对软错误和过程变化的弹性

考虑到芯片堆叠动态随机存取存储器的高可靠性要求，Sim 等人[36]提出了一种在细粒度(例如，单比特故障)和粗粒度(例如，行、存储体和通道故障)级别提供保护的技术。他们注意到，在管芯堆叠的动态随机存取存储器有限责任公司中，在堆叠中添加额外的芯片来存储纠错码信息可能是不实际的，因为纠错码的开销变得非常大，因此，他们仅利用非纠错码堆叠的动态随机存取存储器芯片。他们使用循环冗余校验来代替使用单错误纠正，双错误检测。CRCs 可以检测多位错误，无论这些错误是否聚集在一起，因此，它们大大提高了错误检测能力，并降低了静默数据损坏(SDC)率[64]。他们还注意到，当缓存中的干净数据出现错误时，仅仅检测到错误就足够了，因为可以从主内存中检索到正确的副本。相比之下，对于脏数据，需要纠错和检测，因为缓存中的修改副本是唯一有效的副本[64]。基于此，他们的技术依赖于主存储器来纠正动态随机存取存储器高速缓存中干净块的错误，并使用其他存储体中脏块的复制来为它们提供纠错。为了避免复制的容量开销，他们建议只复制关键应用程序或内存区域，并将脏块从动态随机存取存储器缓存写入主内存。

Zhao 等人[50]注意到，工艺变化可能导致 DRAM LLC 中不同子存储库中的非均匀存取时间。他们提出了通过将数据从慢速子库迁移到快速子库来减少平均访问时间。一种技术总是将数据迁移到最快的子银行。由于这可能会引起最快子库的竞争，第二种技术将一个等级中的子库分成几个层，其中每个层具有几个速度相似的子库。数据从较慢的层迁移到较快的层，在那里数据被均匀地分布到其子库。由于动态随机存取存储器高速缓存提供大容量，几个应用程序的工作集适合快速子库，这使得它们的技术有效。

3.11 缓存重新配置技术

Chang 等人[54]注意到,管芯堆叠的 DRAM 高速缓存提供了大容量,然而,由于它们的工作集尺寸小,几个应用和应用阶段可能不会受益于这种容量。因此,由于刷新开销和外围电路中的泄漏,未使用的高速缓存部分消耗功率。他们提出了一种高速缓存重新配置技术。传统的重新配置方法(例如[26, 37])重新映射存储在通电和断电存储体组中的数据,这导致大量开销。他们的技术使用一致散列算法[54]工作,由此机器故障(或类似地,关闭动态随机存取存储器组)导致仅重新映射映射到该机器(或动态随机存取存储器组)的数据。此外,重新映射是以负载平衡的方式完成的,这不会将通电存储体变成导致高速缓存访问延迟增加的热点,并且已经映射到通电存储体的数据保持在原来的位置。虽然先前的技术(例如[37])允许将高速缓存重新配置为仅二的幂的集合并计数,但是 Chang 等人[54]的技术没有施加这种限制。

3.12 静态随机存取存储器/动态随机存取存储器混合高速缓存

马丹等人[49]提出了一种 3D 混合静态随机存取存储器/动态随机存取存储器高速缓存设计,旨在将静态随机存取存储器的低延迟优势和动态随机存取存储器的高密度优势结合起来。他们假设一个具有三层的堆叠式处理器,其中底层包含 16 个内核,中间层包含 16 个静态随机存取存储器组,顶层包含 16 个动态随机存取存储器组。静态随机存取存储器和动态随机存取存储器一起构成 L2 高速缓存。基于离线分析,他们首先确定每个内核的私有页面和多个内核共享的页面。然后,使用操作系统页面着色,他们将私有页面放在核心之上的静态随机存取存储器组中,并将共享页面放在一个中央静态随机存取存储器组中。由于运行在不同内核上的应用程序呈现不同的 L2 缓存需求,因此将应用程序的页面溢出到相邻存储体中有助于调整每个内核的缓存配额,但是,这也会增加访问延迟和存储体间网络的压力。为了避免这种情况,他们的技术规范将额外的页面溢出到第三维度,即在静态随机存取存储器高速缓存体正上方的动态随机存取存储器体。由于与静态随机存取存储器相比,动态随机存取存储器的密度是 8,所以动态随机存取存储器组的激活将高速缓存容量从 1MB 增加到 9MB。此外,通过尝试服务来自静态随机存取存储器库的大多数请求,他们的技术保持平均高速缓存访问延迟小。至于高速缓存重新配置方法,等人使用了选择性通路方法,而 Chang 等人[54]使用了选择性集合方法[26]。

Inoue 等人[43]提出了一种静态随机存取存储器/动态随机存取存储器混合高速缓存结构。他们假设处理器设计有与核心层在同一层的小型静态随机存取存储器 L2 高速缓存(注意与马丹等人[49]的设计的区别,其中静态随机存取存储器高速缓存在单独的层)和大型管芯堆叠的动态随机存取存储器 L2 高速缓存。使用性能分析,可以估计应用程序的缓存需求,并基于此,将缓存配置为在应用程序执行之前以以下两种模式之一工作。当高速缓存需求较小时,动态随机存取存储器高速缓存是电源门控的,并且仅使用静态随机存取存储器高速缓存,因此,访问延迟保持较小。对于高速缓存需求较大的应用,静态随机存取存储器和动态随机存取存储器都保持活动状态,静态随机存取存储器用于存储动态随机存取存储器高速缓存中存储的数据的标签。在这种模式下,容量变得更高,代价是动态随机存取存储器的存取延迟更高。因此,在同一时间,静态随机存取存储器或动态随机存取存储器,而不是两者都存储数据,这使得他们的技术不同于马丹等人[49],其中静态随机存取存储器和动态随机存取存储器可以同时存储数据。

Hameed 等人[44]提出了一种混合静态随机存取存储器-动态随机存取存储器 LLC (L3),与使用 L3 静态随机存取存储器和 L4 动态随机存取存储器高速缓存的层次结构相比,它避免了数据重复。他们指出,在共享有限责任公司中,来自抖动应用程序的不必要的填充请求会延迟来自非抖动应用程序的关键读/写访问,从而产生内核间动态随机存取存储器干扰。为了解决这个问题,他们提出了一个策略来动态决定从主存储器中取出的一行是放在有限责任公司的静态随机存取存储器和动态随机存取存储器部分,还是只放在静态随机存取存储器部分。他们的技术根据重用距离检测应用程序是否抖动,对于这种应用程序,该行被放置在动态随机存取存储器高速缓存中的概率很低。这避免了对动态随机存取存储器的不必要的插入。与 Inoue 等人[43]不同的是,他们并不是为了节能而对动态随机存取存储器进行功率门控,而是专注于减少动态随机存取存储器中的内核间干扰以提高性能。

3.13 动态随机存取存储器的性能模型

Gulur 等人[22]为静态随机存取存储器中的标签和动态随机存取存储器中的标签组织提出了动态随机存取存储器高速缓存的分析性能模型。它们的模型考虑了关键参数,如动态随机存取存储器缓存和片外存储器时序值、缓存块大小、标签缓存/预测器的命中率和应用特性。他们的模型估计了 DRAM 缓存的平均访问延迟和请求到达率。基于他们的模型,他们表明,为了使动态随机存取存储器中的标签设计优于静态随机存取存储器中的标签设计,辅助标签高速缓存/预测器的命中率需要非常高。他们还注意到,当动态随机存取存储器缓存命中率高时,动态随机存取存储器缓存的大量流量会导致争用和排队延迟。在这种情况下,一些缓存流量可以转移到空闲的主内存,以实现负载平衡。使用他们的模型,他们还

提出一种负载平衡方案，通过确定转移到主存储器的访问的最佳比例来最小化平均等待时间。

3.14 与NVM缓存的比较

Dong 等人[56]将高速缓存与SRAM和DRAM高速缓存在面积、性能和能量方面进行了比较。像STT-RAM这样的非易失性存储器消耗接近零的泄漏功率，然而，它们的写入延迟和能量明显高于静态随机存取存储器甚至动态随机存取存储器[20]。STT-RAM的密度接近动态随机存取存储器，与静态随机存取存储器相比是4倍。使用架构模拟，他们表明，由于刷新操作的要求，动态随机存取存储器比STT-RAM消耗更多的功率，而他们的性能值几乎相似。

虽然字节可寻址的非易失性存储器与动态随机存取存储器相比可能表现出更高的能效，但同样值得注意的是，就商业可行性和成熟度而言，动态随机存取存储器优于这些非易失性存储器。此外，非易失性存储器还存在可靠性问题[64]，例如，相变存储器单元的电阻随着时间的推移而增加，因此，多层（多级单元）相变存储器可以开始表示与最初不同的值存储的。因此，4级相变存储器的软错误率可能比动态随机存取存储器高106倍[65]。

此外，非易失性存储器的写入耐久性为或-数量级小于动态随机存取存储器。例如，当动态随机存取存储器的写入耐久性为1016时，相变存储器和随机存取存储器的该值分别为108和1011[20, 38, 66]。对于STT-RAM，该值预计为1015，尽管迄今为止的最佳结果仅显示4 1012[20]。这不仅限制了非易失性存储器件的寿命，而且也是一个关键的问题

恶意攻击者和贪婪用户的安全漏洞，他们可能会在保修期前通过写攻击使系统失败，以获得新系统。与非易失性存储器不同，动态随机存取存储器没有这种写入耐久性问题。

4 未来的挑战和结论

虽然动态随机存取存储器高速缓存可以为功耗问题提供短期解决方案，但即使与在20MW的功耗预算内寻求每秒1018次运算的数据库云服务器计算的目标相比，它们也远远不够[37]。到

为了实现这个目标，需要跨整个软件栈的解决方案。例如，在设备级，新的阵列组织可以减少重新刷新需求和动态随机存取存储器访问能量[67]。在体系结构层面，多种技术如数据压缩、等级子集设置、访问调度等。

[24]可以协同整合，将它们中最好的结合在一起。类似地，编译器和操作系统技术可用于执行数据放置，通过将常用数据放在同一页上来提高命中率[30]。

随着特征尺寸的缩放，工艺变化的影响（影响动态随机存取存储器单元的保持期）和动态随机存取存储器单元对软错误的敏感性也增加了[64]。虽然对动态随机存取存储器高速缓存的初步研究主要集中在架构设计和性能/能源问题上，但展望未来，对其他问题（如弹性、热管理等）的全面评估仍在继续。将被明确要求来确保这些动态随机存取存储器高速缓存的可靠运行。

使用模拟器或真实系统进行的架构研究对于评估为动态随机存取存储器提出的设计创新至关重要。由于其大尺寸，动态随机存取存储器高速缓存需要大的评估窗口来执行代表性的研究。然而，由于现有模拟器的速度较慢，以及存储大量工作负载轨迹的要求，大型缓存的完全设计空间探索可能是不可行的。此外，由于堆叠式动态随机存取存储器高速缓存的新兴特性，它们的真实原型可能无法广泛获得，或者在经济上不可行。显然，堆叠式动态随机存取存储器的制造和商业可行性的改进，以及极快模拟基础设施的发展，将非常有助于推动动态随机存取存储器高速缓存的进一步研究。

存储系统在决定高端计算系统的性能方面起着至关重要的作用，考虑到其他存储技术（如静态随机存取存储器和非易失性存储器）的局限性，动态随机存取存储器有望成为设计千兆字节高速缓存的最合适的选择。在本文中，我们回顾了几种管理动态随机存取存储器缓存的技术。为了强调它们的相似性和差异，我们根据几个关键参数对它们进行了分类。我们还简要讨论了这一领域面临的挑战。希望本文能对芯片设计者、计算机架构师和系统研究人员有所帮助，并推动这一领域的进一步研究。

参考

- [1] 南鲁苏、马尔乔诺、艾尔斯、谭世安、陈文伟、马丁、南李, S. Vora, R. Varada和E. Wang, “Ivytown:22nm 15核企业级至强处理器家族”, 载于IEEE国际固态电路会议(), 2014年, 第102-103页。
- [2] D.米洛耶维奇、伊德贡吉、日夫吉奇、厄泽尔、洛特菲-卡姆兰、A. Panteli, A. Prodromou, C. Nicopoulos, D. Hardy, B. Falsari等, “高能效片上服务器上云工作负载的热特性”, 载于国际计算机设计会议(ICCD), 2012年, 第175-182页。
- [3] 米 (meter 的缩写) 《异构存储器架构:混合芯片堆叠和封装外存储器的硬件/软件方法》, HPCA, 2015年。
- [4] B. 罗杰斯, 克里希纳, 贝尔, 伍克武, 姜 Y. Solihin, “缩放带宽墙:CMP缩放的挑战和途径”, 计算机架构国际研讨会(ISCA), 2009年, 第371-382页。
- [5] 名词 (noun 的缩写) 哈德维拉斯, m. 费尔德曼, a. 艾拉马基和b. 法萨非, “功率扩展:1K核芯片的终极障碍”, 西北大学, 理工学院. NWU众议员-EECS-2010年10月5日。
- [6] B. 布莱克, 安娜瓦拉姆, 布雷克鲍姆, 德瓦尔, 姜, G. 罗、麦考利、莫罗、纳尔逊、潘图索

- 等,“芯片堆叠(3D)微体系结构”,载于国际微体系结构研讨会,2006年,第469-479页。
- [7] T.徐,李杰伯格,滕胡恩,“探索用于3D片上网络架构的DRAM末级高速缓存”,载于《先进材料研究》,第403卷,2012年,第4009-4018页。
- [8] 米(meter的缩写)库雷希和罗赫,“动态随机存取存储器高速缓存架构中的基本延迟权衡:用简单实用的设计胜过不切实际的静态随机存取存储器标签”,载于国际微架构研讨会,2012年,第235-246页。
- [9] “英特尔至强融核骑士处理器将搭载高达16GB的板载堆叠动态随机存取存储器增压混合存储立方体(HMC)。”<http://wccftech.com/intel-xeon-phi-骑士-登陆-处理器-堆叠-dram-hmc-16gb/>,2014。
- [10] C.周,贾列尔和库雷希,“CAMEO:一种具有主存容量和硬件管理高速缓存灵活性的两级存储组织”,载于国际微体系结构研讨会(MICRO),2014年,第1-12页。
- [11] 南尹,李俊杰,刘立军,魏世伟,郭永安,“协作管理末级DRAM缓存中的动态写回和插入策略”,载于《欧洲设计、自动化与测试》(DATE),2015年,第187-192页。
- [12] X.姜,赵,厄普顿,伊耶,马基-内尼,纽厄尔,索利辛,巴拉苏布拉蒙尼,“CHOP:基于自适应滤波器的动态随机存取存储器缓存技术在高性能计算机体系结构国际研讨会(HPCA)上的应用”,2010年,第1-12页。
- [13] J.-金圣贤,吴圣贤,李德贤,黄圣贤,黄圣贤, B.纳,金俊杰,朴海龙,柳俊伟,朴光洙 K.姜,金圣贤,金海辉,姜明邦,赵海辉,张大千, C.韩,李俊斌,崔俊生,“基于堆叠的1.2V 12.8GB/s 2Gb 移动宽I/O DRAM及其4x 128 I/O”,载于《IEEE国际固态电路会议技术论文摘要》,2011年,第496-498页。
- [14] 单位姜、钟海杰、许世华、安世华、李世华、查世华、J.安道权,金俊辉,李俊伟等,“使用穿透硅通孔技术的8Gb 3D DDR3 动态随机存取存储器”,载于《固态电路会议-技术论文摘要》,2009年。ISSCC 2009。IEEE国际,2009,第130-131页。
- [15] 泰扎隆半导体,“用于芯片堆叠应用的八爪鱼8端口动态随机存取存储器”www.tachyonsemi.com/memory/数据表/TSC_10080_x_0_1.pdf,2010年。--
- [16] J. Jeddeloh 和 B. Keeth,“混合存储立方体新DRAM架构提高密度和性能”,在VLSI技术研讨会(VLSIT)上,2012年,第87-88页。
- [17] D. U. Lee, K. W. Kim, K. W. Kim, K. S. Lee, S. J. Byeon, J. H. Kim, J. H. Cho, J. Lee 和 J. H. Chun,“具有有效I/O测试电路的1.2V 8Gb 8通道128GB/s 高带宽存储器(HBM)堆叠DRAM”,IEEE固态电路杂志,第50卷,第1期,第191-203页,2015年。
- [18] <http://www.pcper.com/reviews/General-Tech/High-带宽-内存-HBM-架构-AMD-计划-未来-图形处理器>,2015。
- [19] 英伟达, <http://devblogs.nvidia.com/parallelforall/nvlink-帕斯卡-堆叠-记忆-喂养-食欲-大数据/>,2014。
- [20] 南米塔尔、维特李和李德生,“管理嵌入式动态随机存取存储器和非易失性片上高速缓存的架构方法综述”,《并行和分布式系统上的IEEE事务》,2015年。
- [21] 名词(noun的缩写)Gulur, M. Mehendale, R. Manikantan 和 R. Govindarajan,“双模DRAM缓存:提高命中率、命中延迟和带宽”,载于国际微体系结构研讨会(MICRO),2014年,第38-50页。
- [22] 名词(noun的缩写)古尔、梅亨代尔和戈文达拉扬,“动态随机存取存储器高速缓存的综合分析性能模型”,载于性能工程国际会议,2015年,第157-168页。
- [23] K.陈,李,安,布罗克曼和朱皮,“CACTI-:3D芯片堆叠DRAM主存储器的体系结构建模”,载于《欧洲设计、自动化和测试会议》,2012年,第33-38页。
- [24] 南米塔尔,“动态随机存取存储器电源管理架构技术综述”,《国际高性能系统架构杂志》,第4卷,第2期,第110-119页,2012年。
- [25] G. “多核处理器的3D堆叠内存架构”,载于国际计算机架构研讨会(ISCA),2008年,第453-464页。
- [26] 南米塔尔,“提高高速缓存能效的架构技术调查”,爱思唯尔可持续计算:信息技术与系统,第4卷,第1期,第33-43页,2014年3月。
- [27] F. Hameed, L. Bauer 和 J. Henkel,“通过新颖的标签高速缓存架构减少静态随机存取存储器/动态随机存取存储器高速缓存层次结构中的延迟”,载于设计自动化会议(DAC),2014年,第1-6页。
- [28] X.董,谢,穆利马诺哈尔,朱皮杰,“具有片上存储器控制器支持的简单而有效的异构主存储器”,载于高性能计算、网络、存储和分析国际会议,2010年,第1-11页。
- [29] G. “为超大规模芯片堆叠的动态随机存取存储器高速缓存有效地实现传统的块大小”,载于国际微体系结构研讨会,2011年,第454-464页。
- [30] G.陆,贾亚泽纳,麦格拉思,奥康纳,雷恩哈特和钟,“异质芯片堆叠和片外存储系统中的挑战”,载于In Proc.关于SoCs、异构性和工作负载的第三次研讨会(SHAW),2012年。
- [31] 米(meter的缩写)埃-纳库齐、阿塔、帕帕多普洛斯、泽布丘克、杰格和莫肖沃斯,“一种用于大芯片堆叠动态随机存取存储器的双颗粒命中失误检测器”,载于《欧洲的设计、自动化和测试》,2013年,第89-92页。
- [32] J. “使用精细粒度的动态随机存取存储器高速缓存管理实现高效和可扩展的混合存储器”,《计算机体系结构通讯》,第11卷,第2期,第61-64页,2012年。
- [33] 米(meter的缩写)Poremba等,“DESTINY:一种新兴3D NVM和eDRAM缓存建模工具”,载于《欧洲设计自动化与测试》。IEEE,2015,第1543-1546页。
- [34] C.《蝙蝠侠:最大化混合存储系统的带宽利用率》,佐治亚理工学院。代表TR-CARET-2015-01,2015。
- [35] J. Sim, G. H. Loh, H. Kim, M. O'Connor 和 M. Thottethodi,“一种用于有效命中推测和自我平衡调度的基本干净的DRAM高速缓存”,载于国际微体系结构研讨会(MICRO),2012年,第247-257页。
- [36] J. “弹性芯片堆叠式动态随机存取存储器高速缓存”,载于计算机架构国际研讨会,2013年,第416-427页。
- [37] 南Mittal 和 Z. Zhang,“EnCache:一种用于提高缓存能量效率的基于动态轮廓的重构技术”,《电路、系统与计算机杂志》,第23卷,第10期,2014。
- [38] 南米塔尔和维特李,“AYUSH:延长静态随机存取存储器-非易失性随机存取存储器混合高速缓存寿命的技术”,IEEE计算机体系结构通讯,2015年。
- [39] 长度赵,艾叶,伊利卡尔和纽厄尔,“探索用于化学机械抛光服务器平台的动态随机存取存储器缓存体系结构”,国际计算机设计会议,2007年,第55-62页。
- [40] Y. 通过共享感知替换和流感知插入策略的高效共享高速缓存管理,IEEE并行与分布式处理国际研讨会,2009,第1-11页。
- [41] H.孙,刘俊杰,阿尼贡迪,郑恩正,卢俊杰,肯和 T.张,“3D DRAM的设计及其在3D集成多核计算系统中的应用”,IEEE计算机设计与测试,2013。
- [42] G. “用自适应多队列策略扩展3D堆叠动态随机存取存储器缓存的有效性”,载于国际微体系结构研讨会,2009年,第201-212页。
- [43] K.井上、桥口、上野、福本聪和木井俊,“用于高性能和低功耗的3D实现的静态随机存取存储器/动态随机存取存储器混合高速缓存架构”,在国际中西部电路和系统研讨会上,2011年,第1-4页。
- [44] F. Hameed, L. Bauer 和 J. Henkel,“多核混合静态随机存取存储器和动态随机存取存储器高速缓存层次结构的自适应高速缓存管理”,载于《欧洲设计、自动化和测试会议与展览》(DATE),2013年,第77-82页。
- [45] F. Hameed, L. Bauer 和 J. Henkel,“在DRAM高速缓存中使用自适应存储体映射策略减少内核间高速缓存争用”,载于硬件/软件代码设计和系统综合国际会议(CODES+ ISSS),2013年,第1-8页。
- [46] F. Hameed, L. Bauer 和 J. Henkel,“通过新颖的集合映射策略同时优化 dram 缓存命中延迟和未命中率”,在国际编译器、架构会议上

和《嵌入式系统综合》，2013年，第11页。

- [47] 名词 (noun 的缩写) M. Tshibangu, P. D. Franzon, e. 罗滕博格和 W. R. Davis, “在 Tezzaron 堆叠 DRAM 中映射的 L2 高速缓存的控制器设计”, 在 IEEE 国际 3D 系统集成会议 (3DIC) 上, 2013 年, 第 1-4 页。
- [48] 米 (meter 的缩写) R. Meswani, G. H. Loh, S. Blagodurov, D. Roberts, J. Slice 和 M. Ignatowski, “在 exascale 计算机中实现高效的程序员管理的两级内存层次结构”, 载于《高性能计算 (Co-HPC) 的软硬件协同设计》, 2014 年, 第 9-16 页。
- [49] 名词 (noun 的缩写), 赵, 尤迪皮, 巴拉苏-拉莫尼, 伊耶, 马基内尼和纽厄尔, “在三维堆叠可重构高速缓存体系结构中优化通信和容量”, 载于高性能计算机体系结构国际研讨会, 2009 年, 第 262-274 页。
- [50] B. 赵, 杜, 张, 杨, “芯片堆叠多核处理器中的变异容忍非均匀 3D 缓存管理”, 载于 IEEE/ACM 微体系结构国际研讨会, 2009 年, 第 222-231 页。
- [51] D. Jevdjic, G. H. Loh, C. Kaynak 和 B. Falsafi, “Unison Cache: 一种可扩展且有效的芯片堆叠 DRAM 缓存”, 载于国际微体系结构研讨会 (MICRO), 2014 年, 第 25-37 页。
- [52] C.-C. Huang 和 V. Nagarajan, “ATCache: 通过小型 SRAM 标签缓存降低 DRAM 缓存延迟”, 载于并行架构与编译国际会议, 2014 年, 第 51-60 页。
- [53] D. 服务器的芯片堆叠式动态随机存取存储器: 命中率、延迟还是带宽? 足迹缓存拥有一切”, 载于《计算机体系结构国际研讨会》, 2013 年, 第 404-415 页。
- [54] K. 张, 罗海浩, 汤道迪, 奥康纳, 长度 Subramanian 和 O. Mutlu, “通过硬件一致的散列机制实现大型 DRAM 高速缓存的有效动态调整大小”, 卡内基梅隆大学, 技术。代表 2013-001, 2013。
- [55] K. 朴信贤、朴信贤、黄国伟、申德杰、崔英伟和朴信伟, “MN-MATE 平台的分层和混合主存储器的高效内存管理”, 载于《多内核和多内核编程模型和应用国际研讨会》, 2012 年, 第 83-92 页。
- [56] X. 董, 吴晓霞, 孙光洋, 谢海红, 陈永红, “作为通用存储器替代物的 3D 堆叠磁 RAM() 的电路和微体系结构评估”, 载于设计自动化会议, 2008 年, 第 554-559 页。
- [57] 南李俊杰和郑俊明, “用未来的存储器技术取代静态随机存取存储器高速缓存的混合高速缓存体系结构”, 载于 IEEE 电路与系统国际研讨会, 2012 年, 第 2481-2484 页。
- [58] C. 周, 贾列尔和库雷希, “贝尔: 减轻千兆级动态随机存取存储器高速缓存带宽膨胀的技术”, 在计算机体系结构国际研讨会 (ISCA), 2015 年。
- [59] Z. Jaksic 和 R. Canal, “基于 DRAM 的相干缓存以及如何利用相干协议来降低刷新能量”, 载于《欧洲设计、自动化和测试会议与展览》(DATE), 2014 年, 第 1-4 页。
- [60] W. 云, 郑俊杰, 康金明, “通过的三维堆叠 DRAM 高速缓存的温度感知能量最小化”, 国际 SoC 设计会议 (ISOC), 2012, 第 475-478 页。
- [61] 米 (meter 的缩写) Ghosh 和 h-h. s. Lee, “智能刷新: 一种增强的存储控制器设计, 用于降低传统和 3D 芯片堆叠式动态随机存取存储器的能耗”, 载于国际微体系结构研讨会, 2007 年, 第 134-145 页。
- [62] Y. 潘和张, “使用三维 (3d) DRAM 堆叠提高 VLIW 处理器性能”, 在 IEEE 专用系统、架构和处理器国际会议 (ASAP) 上, 2009 年, 第 38-45 页。
- [63] C. 基于总线的多处理器的子块高速缓存一致性协议的设计和评估。94-05-02, 1994 年。
- [64] 南米塔尔和维特尔, “计算系统建模和提高可靠性技术的调查”, 并行和分布式系统的 IEEE 交易 (TPDS), 2015 年。
- [65] 名词 (noun 的缩写) 三级单元相变存储器: 迈向高效可靠的存储系统, 2013 年, 第 440-451 页。
- [66] 南米塔尔, “相变存储器电源管理技术调查”, 《国际计算机辅助工程与技术杂志》, 2014 年。
- [67] B. 吉里达尔、西斯拉克、杜加尔、德累斯顿斯基、陈, R. 帕蒂, b. 霍尔德, c. 查克拉巴蒂, t. 穆奇和 d. 布劳, “探索动态随机存取存储器组织以获得高能效和高可靠性的超大规模存储器”, 载于 Int. 糖膏剂《论高性能计算、网络、存储和分析》, 2013 年, 第 23 页。

斯巴塞·米塔尔在印度 IIT 获得电子和通信工程学士学位, 在美国爱荷华州立大学获得计算机工程博士学位。他目前在 ORNL 大学担任博士后研究助理。他的研究兴趣包括高速缓存和主存储器系统、弹性和能效、非易失性存储器和 GPU 架构。

杰弗里·维特尔博士是 ORNL 和佐治亚理工学院的联合任命。在 ORNL, 他是著名的 R&D 员工, 也是未来科技集团的创始人。在 GT, 他是计算科学与工程学院的联合教授, NSF Track 2D 实验计算设施的项目总监, 该设施使用图形处理器进行大规模异构计算, 并且是 NVIDIA CUDA 卓越中心的指导者。他的研究兴趣包括大规模多线程处理器、极端规模系统的内存架构和异构多核处理器。