arXiv:2304.05442v1 [cs.AR] 11 Apr 2023

非对称多核处理器中分区缓存的性能研究

Murali Dadi、Shubhang Pandey、Aparna Behera 和 T G Venkatesh

*印度理工学院马德拉斯分校电气工程系，印度金奈。

*通讯作者。电子邮箱：ee19s057@smail.iitm.ac.in；tgvenky@ee.iitm.ac.in；贡献作者：muralidadi963@gmail.com；ee16d038@smail.iitm.ac.in；

摘要

当前的工作负载和应用程序高度多样化，面临着功耗墙和内存墙问题等严峻挑战。人们已针对多级缓存发展出不同策略，以缓解这些问题。此外，非对称多核处理器（AMP）是应对这类多样化应用的一种可行方案。本文研究了不同缓存分区下，二级缓存和末级缓存针对各种非对称多核处理器配置的性能。此外，本研究还针对 PARSEC 和 SPLASH2 基准测试套件中的一组多线程基准测试，在中等规模输入条件下探究了最优缓存分区方案。我们研究了块替换策略的效果及其对总片上功耗、二级缓存和末级缓存未命中率等关键指标的影响。本研究提出了一种介于二级缓存和末级缓存完全共享与完全私有两种极端方案之间的非对称多处理器中间缓存设计，有助于实现理想的功耗水平和最优的缓存未命中代价。

关键词：非对称多核处理器、二级缓存、末级缓存、缓存替换策略、CPU功耗

1 引言

当前的工作负载和应用高度多样化，面临着功耗墙和内存墙问题等严峻挑战。为缓解这些问题，人们针对缓存的多个层级提出了不同策略。此外，非对称多核处理器（AMP）为应对这类多样化应用提供了一种可行方案。本文研究了不同非对称多核处理器配置下，各种缓存分区对二级缓存和末级缓存性能的影响。此外，我们还针对 PARSEC 和 SPLASH2 基准测试套件中的一组多线程基准测试，研究了中等规模输入下的最佳缓存分区方案。我们研究了块替换策略的影响，以及其对总片上功耗、二级缓存和末级缓存未命中率等关键指标的影响。本研究提出了一种介于二级缓存和末级缓存完全共享与完全私有两种极端方案之间的非对称多处理器中间缓存设计，有助于实现预期的功耗值并优化缓存未命中代价。

近年来，在设计高性能、能效高的非对称多核处理器方面取得了巨大进展。然而，性能与功耗之间的权衡在处理器设计中起着至关重要的作用。随着集成电路技术的快速发展，单位面积内的晶体管数量不断增加，导致功率密度（W/cm^2）上升 [1]。此外，目标函数（更高性能或更低功耗）也可能因设备的需求和运行条件而有所不同。例如，在移动设备中，闲置期间需要优化能耗以延长电池续航时间，而在活跃期间则需要优先考虑性能。末级缓存是处理器资源之一，会显著影响系统性能和能耗。因此，如何高效管理末级缓存成为一项重大挑战 [1],[2]。基于这一点，本文旨在对非对称多处理器的末级缓存进行全面的性能评估，具体如下文所述。请注意，在本文的语境中，我们采用三级缓存，并将 L3 和末级缓存这两个术语互换使用。

本文研究了非对称多核处理器体系结构中末级缓存的性能。本文的概要如下。在多核体系结构中，管理共享末级缓存是一项关键任务。因此，我们探讨了二级缓存和末级缓存的不同配置对多核体系结构关键系统指标的影响，包括二级缓存未命中率、L3 缓存未命中率和总功耗。此外，我们还研究了不同核心（具有不同工作频率）以及不同执行顺序（顺序执行或乱序执行）对非对称多核处理器功耗的影响。这构成了本文的概要。

本文其余部分组织如下：第2 节简要综述了末级缓存性能评估方面的相关文献。随后，第3节介绍了与末级缓存相关的研究及相应的仿真结果。

根据这些仿真结果得出的推论以及结论性评述将在第4节中介绍。

2 文献综述

本节回顾了与末级缓存性能和能效研究相关的现有文献。微处理器中经常采用缓存存储器来提升系统性能，因此这些缓存一直是众多研究的主题[2–6]。末级缓存的替换策略会显著影响片外未命中流量和功耗。Peneau 等人研究了不同的末级缓存替换策略如何影响系统性能和能耗 [7]。异构多核架构需求旺盛，而现有替换策略在应用于异构多核系统时面临重大挑战。Ramtake 等人研究了多核系统中相联度对一级缓存、二级缓存的影响，考察了缓存命中率和 IPC（每周期指令数） [8]。现代 VLSI 芯片在处理器上集成了更大的缓存，而管理如此大的缓存容量会产生相当大的开销。Wu 等人为多核处理器芯片中的共享末级缓存提出了一种基于机器学习的管理方案 [9]。Jang 等人 [10] 提出了一种适用于更大末级缓存的设计，即使在高粒度下也能实现良好性能。Anandkumar 等人为异构多核处理器提出了一种新的混合缓存替换策略，将最近最少使用和最不经常使用替换策略结合起来[11]。多核系统中的异构性可以通过改变各个核心的频率、缓存大小及其他缓存参数来实现。Silva 等人研究了 HMP（异构多核处理器）采用不同缓存大小的优势，以及调度技术如何利用这些优势来降低总体未命中率 [12]。现代多核处理器芯片（CMP）中的末级缓存通常由所有核心共享。处理器更频繁地使用共享缓存，因此共享数据被逐出会导致更多缓存未命中[13]。因此，为了有效利用多核处理器芯片中的共享末级缓存，Sato 等人提出了缓存分区，以减少不必要的逐出并保护共享数据 [14]。该方法将共享数据与私有数据分开，并通过缓存分区为两类数据分别分配缓存空间。已有多项研究针对共享末级缓存分区展开，以提升系统性能，但这些研究都忽略了不同应用在空间局部性方面的异构性。Gupta 等人 [15] 展示了如何利用空间局部性显著提高缓存共享效率。他们指出，采用较大的块大小时，许多内存密集型应用的缓存容量需求可以大幅降低，从而有效地为其他工作负载分配更多容量。在多核处理器芯片（CMP）中，私有末级缓存的访问延迟低于共享缓存。但私有缓存数量增加会导致共享数据重复存储，使缓存总有效容量得不到充分利用，从而降低整体命中率。Chen 等人关于私有缓存和共享缓存的部分研究

缓存阐述了上述问题，包括性能和能耗方面的问题 [16]。为解决上述问题，Yuan 等人提出了一种新的缓存管理技术，通过使用私有末级缓存来提升多核处理器芯片的性能 [17]。Sibai 等人讨论了同构多核架构中共享和私有化二级、三级缓存存储器相关的问题 [18]。

尽管缓存可以显著提升系统性能，但其会消耗大量系统总功耗。Chakraborty 等人 [19] 分析了末级缓存对芯片温度的影响，并提出了一种新策略，可在运行时调整片上末级缓存的大小，从而降低漏电功耗。

从上述文献研究中，我们观察到，关于多核处理器的末级缓存性能已经得到了深入研究。然而，同时共享/划分二级缓存和末级缓存的综合影响，以及替换的影响，仍需进一步研究。最后，在非对称多核处理器背景下，末级缓存的性能尚未受到足够关注。为填补这一空白，我们对末级缓存进行了广泛的性能研究，主要关注处理器的异构性。

本文的独特之处如下。

1. [18] 评估了共享/私有末级缓存对同构多核处理器的性能影响，而我们研究的是同一问题，但针对异构多核架构。2. 在 [12],等大多数论文中，多核处理器的异构性是通过改变缓存大小或块大小引入的。我们论文的一个独特之处在于，除了改变核心频率之外，我们还通过改变执行顺序（顺序执行或乱序执行）来引入异构性。3. 我们工作的另一个创新之处在于，我们全面研究了不同配置的非对称多处理器的性能，这些配置的差异体现在二级缓存和末级缓存的组织方式上。

p分区后得出 p到最佳配置g。

3 非对称多处理器体系结构中的性能权衡

在本节中，我们研究非对称多核体系结构中的性能权衡。考虑到性能和功耗都是首要关注点，我们实现了九种不同的配置。在配置 1 至 5 中，所有核心都是采用乱序执行、运行频率为 2.66GHz 的高性能核心。而在配置 6 至 9 中，我们通过改变核心的执行顺序（顺序执行或乱序执行）以及各核心的运行频率（1GHz 或 2.66GHz），引入了非对称性。在这些配置中，核心 $0-7$ 是以 $1GHz$ 运行的顺序执行核心，而$8-15$ 是以 2.66GHz 运行的乱序执行核心。当乱序执行核心和顺序执行核心共享缓存时，乱序执行核心可能会

表 1：不同配置下的二级缓存和三级缓存详情

配置索引	$L2$ 缓存详情	$L3$ 缓存详细信息	核心频率
配置 1	All 16 核心 共享 $2048KB$ 的 $L2$ 缓存	All 16 核心 共享 $8192KB$ 的 $L3$ 缓存	全部 16 个核心都是 Out-乱序, 运行 at $2.66GHz$
配置 2	Two sets of 8 核心 and each 设置共享 $1024KB$ 的 $L2$ 缓存	all 核心 共享 $8192KB$ 的 $L3$ 缓存	All 核心 are Out-乱序, 运行中的 at 2.66GHz
配置 3	4 个由 4 个核心组成的组，以及每个集合共享 $512KB$ 的 $L2$ 缓存	all 核心 共享 $8192KB$ 的 $L3$ 缓存	All 核心 are Out-乱序, 运行中 at 2.66GHz
配置 4	8 组 2 个核心，且每组共享 $256KB$ $L2$ 缓存	all 核心 共享 $8192KB$ 缓存 $L3$	All 核心 are Out-乱序, 运行 at $2.66GHz$
配置 5	每个核心都配有私有 $L2$ 缓存，容量为 $128KB$	all 核心 共享 $8192KB$ 的 $L3$ 缓存	All 核心 are Out-乱序, 正在运行 at 2。66GHz
配置 6	每个核心都拥有私有 $L2$大小为 $128KB$的缓存	两组各包含 8 个核心以及每个集合共享缓存的大小 $4096KB$	Core $0-7$, 顺序执行, 速度 $1GHz$。核心 $8-15$ 乱序执行，速度 $2.66GHz$
配置 7	每个核心配备私有 $L2$大小的缓存 $128KB$	由 4 个核心组成的四组，并且每组共享大小为 $2048KB$的缓存	核心 $0-7$, 顺序执行, 速度 $1GHz$。核心 $8-15$ 乱序执行，速度 $2.66GHz$
配置 8	每个核心都具有私有 $L2$大小的缓存 $128KB$	8 组 2 个核心，以及每个组共享的缓存 亚洲欧美大小 $1024KB$	核心 $0-7$ 顺序执行, 速度 $1GHz$。核心 $8-15$ 乱序执行，速度 $2.66GHz$
配置 9	每个核心都有私有 $L2$大小为 $128KB$的缓存	每个核心配备 私有 大小为 $512KB$ 的$L3$ 缓存	核心 $0-7$ 顺序执行, 速度 $1GHz$，核心 $8-15$ 乱序执行，速度 $2.66GHz$

占用缓存的很大一部分，这会进一步降低以较低频率运行的顺序执行核心的性能。因此，我们逐步将末级缓存划分给各个核心，从完全共享到完全私有。所有九种配置的其余详细信息见表 1。我们研究所采用的体系结构以及相应的仿真设置如下所示。

3.1 Architecture Studied

我们采用了 Nehalem 体系结构 [20], ，这是由英特尔推出的最成功的处理器体系结构之一。Nehalem 可扩展支持 1 到 16（或更多）个线程以及 1 到 8（或更多）个核心，并具有可扩展、可配置的系统互连，还集成了内存控制器。如图1 所示，该微体系结构采用三级缓存层次结构，其中 $L1$ 缓存容量为 64KB，包括 32KB 数据缓存和 32KB 指令缓存。此外，每个核心还配有 256KB 的 $L2$ 缓存（私有缓存），用于处理数据和指令。最后，它还具有一个容量为 8MB、完全包含且完全共享的末级缓存，所有应用程序均可使用整个缓存。Nehalem 具有更大的乱序执行窗口和调度器，有助于识别更多可并行运行的独立操作。它的核心中还配备了更大的缓冲区，以确保缓冲区不会限制性能。

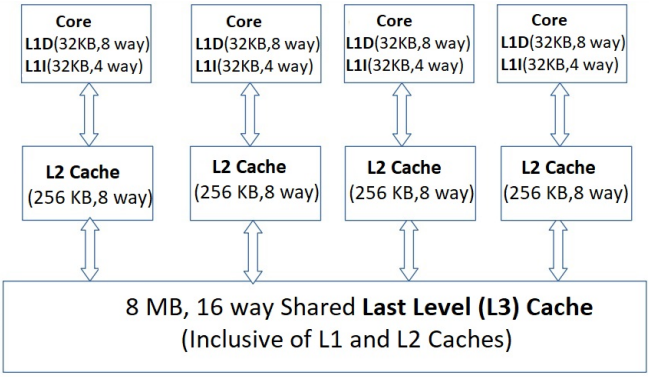


图 1：具有 4 个核心的 Nehalem 体系结构缓存层次结构

3.2 仿真器和所用工作负载

本研究使用了 *Sniper* v7.3 模拟器 [21]。它是一款精确的高速 $X86$ 模拟器，适用于探索不同的异构多核架构。该模拟器还可为多线程、多程序工作负载以及共享内存应用提供高速时序仿真。本研究从 PARSEC 基准测试套件 [22] 和 SPLASH2 基准测试套件[23]中选取了五种不同的工作负载，分别是 PARSEC-Bodytrack、PARSEC-freqmine、SPLASH2-barnes、PARSEC-fluidanimate 和 SPLASH2-radiosity。此外，*McPAT* （多核功耗、面积和时序）框架与 Sniper 集成，用于对众核架构的功耗和面积进行建模。本研究使用 McPAT v1。0 [24] 来获取处理器的不同功耗值。

3.3 仿真结果

在初始阶段，我们对表1中提到的全部九种不同配置进行了仿真。我们使用 L2 缓存未命中率、L3 缓存未命中率和总功耗作为比较指标，对这些配置进行了比较。汇总结果列于表3、4 中，并分别展示在图2 的子图 (a)、(b) 和 (c) 中，以便更好地理解。除 $L2$ 和 $L3$ 缓存级别外，所有配置中的其余参数均相同，具体如表2

从图 2a可以观察到，从配置 1 到配置 5，$L2$ 缓存正从所有核心完全共享逐渐转变为每个核心完全私有。私有缓存越多，发生一致性未命中的可能性就越大，从而提高未命中率。因此 $L2$ 当

表 2：配置详情

核心数量	16
块大小	64 字节
缓存一致性协议	MESI
替换策略	LRU
$L1$ 大小	$32KB(I)$、 $32KB(D)$
$L1$ 共享核心	1（每个核心私有）
$L1$ 相联度	$8(D), 4(I)$
$L2$ 大小	$2048KB$ （总计）
$L2$ 相联度	8
$L3$ 大小	$8192KB$ (总计)
$L3$ 相联度	16

从完全共享转变为完全私有缓存时。不过，从配置 5 到配置 9， $L2$ 缓存对每个核心都是私有的，此时未命中率几乎相同。

表3</style】： $L2$ 缓存未命中率（所有配置）

配置索引	总未命中次数 / 总 $L2$ 访问	$L2$ Miss rate
配置 1	1, 775, 452/32,658, 514	5.43
配置 2	2, 414, 712/23,034, 264	10.48
配置 3	3, 464, 822/24,173, 825	14.33
配置 4	5, 413, 832/20,838, 462	25.98
配置 5	17, 851, 930/20, 985, 015	85.06
配置 6	17, 945, 030/20, 985, 015	85.36
配置 7	18, 105, 468/21, 045, 636	85.99
配置 8	18, 096, 468/21, 045, 663	85.99
配置 9	18, 013, 786/20, 914, 704	86.13

表 4： $L3$ 缓存未命中率及所有配置的功耗

配置索引	总计 $L3$未命中次数/总计 $L3$ 访问次数	$L3$ Miss 比率（%）	总计总功耗（W）
配置 1	83, 872/1, 792,716	4.67	275.238
配置 2	87, 401/2, 430,864	3.59	275.449
配置 3	87, 561/3, 481,529	2.51	274.293
配置 4	82, 067/5, 435, 954	1.51	273.325
配置 5	85, 948/17, 871, 357	0.48	273.718
配置 6	169, 125/17, 984, 595	0.94	263.471
配置 7	309, 358/18,176, 373	1.7	265.63
配置 8	527, 606/18, 178, 885	2.9	268.418
配置 9	1, 033, 091/18, 161, 730	5.688	270.978

从图 2b中可以观察到，当配置从 1 变为 5 时，尽管所有这些情况下三级缓存都是共享的，L3 缓存未命中率仍在下降。如果观察表 4中配置 1 至 5 的三级缓存总未命中次数，可以发现它们几乎相同。由于二级缓存未命中率的增加，对三级缓存的访问次数增加，从而降低了总体 L3 缓存未命中率。在配置 5 至 9 中，三级缓存正从完全共享缓存变为完全私有缓存，由于数据不一致性（相干性未命中）增加，未命中率也随之上升。此外，当私有

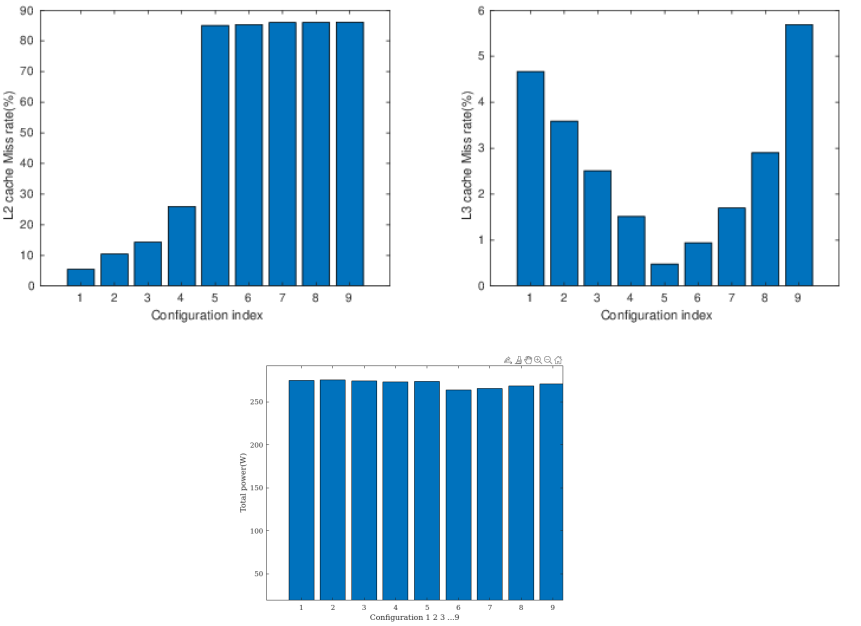


图 2：异构多核架构中的性能权衡 e

缓存的数量增加时，可能会复制相同的数据，这也会因网络缓存容量利用效率低下而增加总未命中率。

从图 2c可以看出，在配置 1 到 5 中，所有核心均以 2.66GHz（乱序执行）运行，因此所有情况下的总功耗几乎相同。但配置 6 到 9 中，有八个核心以 1GHz（顺序执行）运行，另外八个核心以 2.66GHz（乱序执行）运行，因此功耗较低。因此，采用非对称多核后，功耗会降低。不过，从配置 5 到 9，随着 L3 缓存未命中率上升，功耗也随之增加，因为片外主存访问会消耗大量功耗。

3.3.1 配置 1，6，9：采用不同替换策略和不同工作负载

在 3.3 节中，我们发现非对称多核体系结构的 L3 缓存未命中率与总功耗之间存在权衡。我们采用配置 1、配置 6 和配置 9，使用五种不同的工作负载（PARSEC-bodytrack、PARSEC-freqmine、SPLASH2-barnes、PARSEC-fluidanimate、SPLASH2-radiosity），并通过三种不同的替换策略（最近最少使用、最近最常使用、轮询）进一步研究这一权衡。我们考察了 L2 缓存未命中率、L3 缓存未命中率和总功耗

在上述五种不同工作负载下的配置 1、6、9 中。上述仿真采用三种不同替换策略所得的相应结果如图 3 （最近最少使用策略）、图 4 （最近最常使用策略）和图 5 （轮询策略）所示。

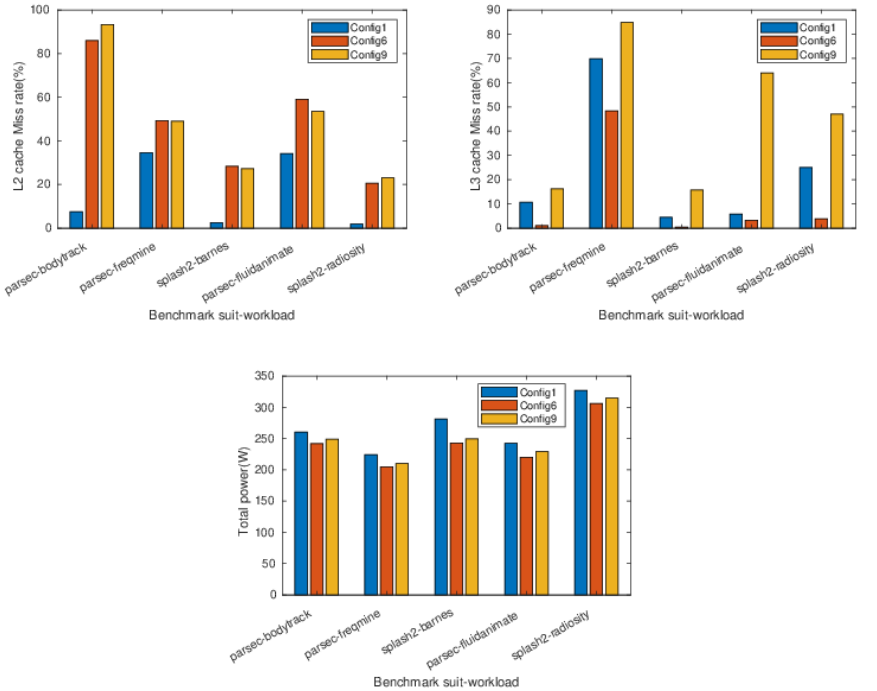


图 3：配置 1，6，9 采用最近最少使用替换策略

无论从三级缓存未命中率还是功耗来看，配置 6 都更优。从上图可以看出，三级缓存未命中率与总功耗之间存在权衡。无论采用何种替换策略或工作负载，从配置 1 变为配置 9 时，所有情况下总功耗都会显著降低，但 $L3$ 未命中率会升高。与最近最常使用策略相比，最近最少使用和轮询替换策略的二级缓存未命中率更低。但我们需要关注三级缓存未命中，而非二级缓存未命中，因为三级缓存未命中会访问主存，从而增加总体延迟和能耗。

对于最近最少使用替换策略，当配置从 1 更改为 6 时，L3 缓存未命中率和功耗分别降低了 11。6% 和 9。058%。此外，当配置从 6 更改为 9 时，L3 缓存未命中率和功耗分别增加了27。25% 和3。06%。当配置从 1 更改为 9 时，L3 缓存未命中率增加

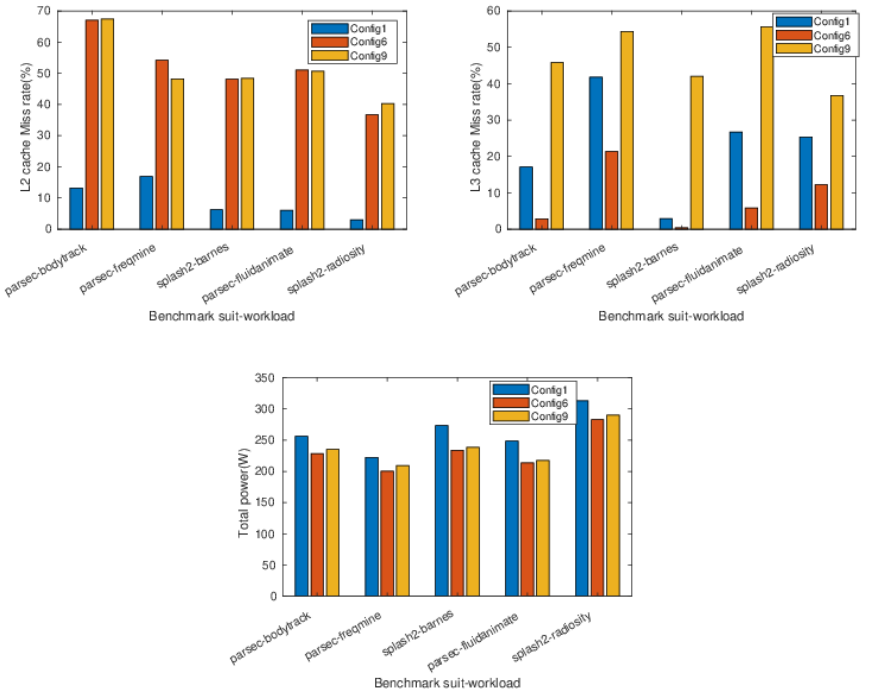


图 4：配置 1，6，9，采用最近最常使用替换策略

降低了13.25%，功耗降低了6.28%。在最近最常使用替换策略中，从配置 1 到配置 6，L3 缓存未命中率降低了14.2%，功耗降低了11.752%。随后从配置 6 到配置 9，L3 缓存未命中率和功耗分别增加了32%和2.75%。从配置1到配置9，L3 缓存未命中率增加了20.25%，功耗降低了9.3%。在轮询替换策略中，从配置 1 到配置 6，$L3$ 未命中率降低了7.6%，功耗降低了10.81%。从配置 6 到配置 9，L3 缓存未命中率增加了28%，功耗增加了3.668%。从配置-置 1 到配置 9，L3 缓存未命中率增加了 20.4%，功耗降低了 7.704%。

4 推断与结论

本文针对非对称多核体系结构，利用其异构性对共享末级缓存进行了性能研究。我们通过改变缓存大小、相联度等参数，并结合不同的替换策略，首先研究了显著影响二级缓存和末级缓存的指标。我们认识到，替换策略对整体缓存性能具有

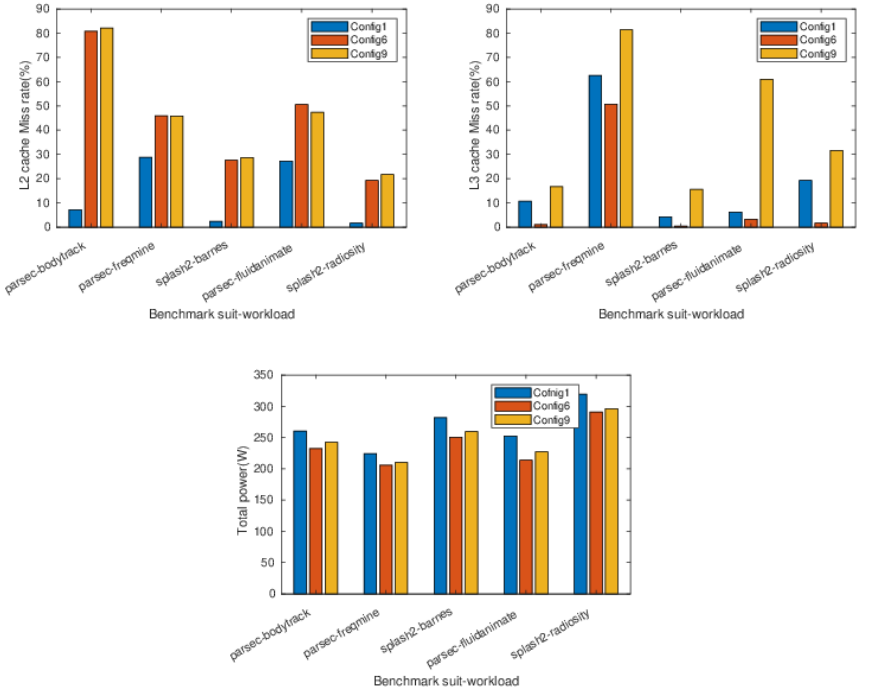


图 5：配置 1，6，9 采用轮询替换策略

显著影响。在研究的第一部分中，我们观察到，增大缓存大小和相联度可以提高缓存的命中率。但我们发现，命中率的提升并非在所有情况下都相同。因此可以说，缓存命中率不仅取决于缓存大小、相联度、块大小等参数，还取决于所采用的替换策略，以及给定工作负载所遵循的不同内存访问模式。在研究的第二部分中，我们

表5：所有配置的比较结果

配置	*L3* 未命中率	功耗
配置 1 到 6	11.3%（下降）	10.54%（下降）
配置 6 到 9	29.08%（增加）	3.154%（增加）
配置 1 到 9	17.9%（增加）	7.761%（下降）

在这项工作中，我们理解了缓存层次结构，并模拟了非对称多核处理器体系结构。我们通过改变各个核心的工作频率和执行方式，将异构性引入我们的 16 核非对称多核处理器中，也就是说，核心采用顺序执行还是乱序执行。需要事先指出的是，在我们对非对称多处理器及其缓存性能进行的整个研究过程中，DVFS

（动态电压与频率调节）保持启用状态。我们观察到，结果中的功耗变化不能按绝对值来考虑；相反，应将其视为相对于各个体系结构的变化。我们在所有核心以相同频率运行的情况下开始仿真，其中每个核心共享二级缓存和LL缓存，并且还采用乱序执行。随后，我们逐步对缓存进行分区，使其成为每个核心的私有缓存，并简化部分核心，例如降低频率值并使其采用顺序执行，以节省功耗。我们的研究表明，配置 6是最佳选择，因为与配置1相比，它可以节省功耗，并且具有最佳的LL缺失率。正如我们所知，片外存储器访问通常比片上存储器访问消耗更多延迟。

我们在多线程 PARSEC 和 SPLASH2 基准测试上评估了我们的体系结构，以了解其在现实场景中的表现。从第 3.3.1 节可以看出，平均而言，从配置 1 到配置 9，，未命中率增加了 17 $L3$ ，功耗降低了 7。761%。我们进一步观察到，从配置 1 到配置 6， $L3$未命中率和功耗分别降低了 11。14% 和 10。54%。相比之下，从配置 6 到配置 9，L3 缓存未命中率增加了 17。9%，功耗增加了 3。154%。因此，配置 6 在性能和功耗方面都表现更佳。上述所有配置的对比结果列于表 5。

我们工作的下一步可以是增加核心数量，并提高系统中的异构性，例如采用灵活的缓存设计、调整重排序缓冲区大小，并提出更多优化方案。

资助： 本研究未获得公共、商业或非营利部门任何资助机构的专项资助。

知情同意： 本文不适用。

利益冲突： 通讯作者代表全体作者声明，不存在利益冲突。

数据可用性声明： 研究期间生成或分析的所有数据

y 均已收录于本文中。

作者贡献：

MuraliDadi：研究的构思与设计、数据分析和/或解释、撰写——初稿、撰写——审阅与编辑。ShubhangPandey：研究的构思与设计、数据分析和/或解释、撰写——初稿、撰写——审阅与编辑。AparnaBehera：研究的构思与设计、数据分析和/或解释、撰写——初稿、撰写——审阅与编辑。TGVenkatesh：研究的构思与设计、数据分析和/或解释、撰写——初稿、撰写——审阅与编辑。

参考文献

[1] Martin，C.：《多核处理器：挑战、机遇与新兴趋势》

[2] Jamet, A.V.、Alvarez, L.、Jimenez, D.A.、Casas, M.：表征末级缓存替换策略对大数据工作负载的影响。2020年 IEEE 工作负载表征国际研讨会（IISWC）（2020年）。https://doi.org/10.1109/iiswc50251.2020.00022[3] Kharbutli, M.、Solihin, Y.：基于计数器的缓存替换与绕过算法。IEEE Transactions on Computers 57(4)，433–447（2008年）。https://doi.org/10.1109/tc.2007.70816[4]Lin, I.-C.、Chiou, J.-N.：采用缓存分区和访问感知策略的 CMP 体系结构高耐久性混合缓存设计。IEEE Transactions on Very Large Scale Integration (VLSI) Systems 23(10)，2149–2161（2015年）。https://doi.org/10.1109/tvlsi.2014.2361150[5] Ferrer´on, A.、Alastruey-Bened´e, J.、Gracia, D.S.、Arnal, T.M.、Mar´ın, P.I.、Y´ufera, V.V.：一种用于超低电压运行 CMP 的容错末级缓存。Journal of Parallel and Distributed Computing 125，31–44（2019年）。https://doi.org/10.1016/j.jpdc.2018.10.010[6] Montaner, H.、Silla, F.、Fr¨oning, H.、Duato, J.：集群内存分配的新自由度。Cluster Co mputing 15(2)，101–123（2012年）[7] Peneau, P.-Y.、Novo, D.、Bruguier, F.、Sassatelli, G.、Gamatie, A.：末级缓存替换策略的性能与能耗评估。2017年第一届嵌入式与分布式系统国际会议（EDiS）（2017年）。https://doi.org/10.1109/edis.2017.8284032[8]Ramtake, D.、Singh, N.、Kumar, S.、Patle, V.K.：多核系统的缓存相联度分析。2020年计算机科学、工程与应用国际会议（ICCSEA）（2020年）。https: // doi.org/10.1109/iccsea49143.2020.9132884[9] Wu, Q.、Ji, Z.：用于管理共享末级缓存的基于感知器的复制方案。Microprocessors and Microsystems，104310（2021年）。https://doi.org/10.1016/j.micpro.2021.104310

[10] Jang, G., Gaudiot, J.-L.：数据引导：面向大规模芯片的末级缓存设计。2019 年 IEEE 第 21 届高性能计算与通信国际会议、IEEE 第 17 届智慧城市国际会议、IEEE 第 5 届数据科学与系统国际会议（HPCC/SmartCity/DSS）（2019）。https://doi.org/10.1109/hpcc/smartcity/dss.2019.00265

[11] Anandkumar, K.m., Akash, S., Ganesh, D., Christy, M.S.：异构多核心的混合缓存替换策略。2014年计算、通信与信息学进展国际会议（ICACCI）（2014年）。https://doi.org/10.1109/icacci.2014.6968209

[12] Silva, B.D.A.、Cuminato, L.A.、Bonato, V.：使用不同缓存大小降低异构多核处理器的总体缓存未命中率。《2012 年可重构计算与 FPGA 国际会议》（2012 年）。https://doi.org/10.1109/reconfig.2012.6416783[13] Ling, M.、Lu, X.、Wang, G.、Ge, J.：考虑数据共享与一致性的多核共享缓存行为分析建模。《IEEE Access》**9**，17728–17743（2021 年）。https://doi.org/10.1109/access.2021.3053350[14] Sato, M.、Nishimura, S.、Egawa, R.、Takizawa, H.、Kobayashi, H.：保护共享数据的多核处理器缓存分区机制。《2016 IEEE 低功耗与高速芯片研讨会》（COOL CHIPS XIX）（2016 年）。https://doi.org/10.1109/coolchips.2016.7503674[15] Gupta, S.、Zhou, H.：面向空间局部性的缓存分区，实现高效缓存共享。《2015 年第 44 届并行处理国际会议》（2015 年）。https://doi.org/10.1109/icpp.2015.24[16] Chen, C.-W.、Hsia, A.、Zhan, Y.-W.、Liu, T.-J.：面向多核处理器的节能型混合一致性协议。《Cluster Computing》**21**（3），1521–1541（2018 年）[17]Yuan, F.、Ji, Z.：面向具有私有末级缓存的多核处理器的复制感知缓存管理。《2016 年第 2 届 IEEE 计算机与通信国际会议》（ICCC）（2016 年）。https://doi.org/10.1109/compcomm.2016.7925214[18] Sibai, F.：同构多核架构中共享和私有化二级及三级缓存的性能优势。《Microprocessors and Microsystems》**32**，405–412（2008 年）。https://doi.org/10.1016/j.micpro.2008.06.002[19] Chakraborty, S.、Kapoor, H.K.：分析末级缓存对控制芯片温度的作用。《IEEE Transactions on Sustainable Computing》**3**（4），289–305（2018 年）。https://doi.org/10.1109/tsusc.2018.2823542[20] Singhal, R.：《Inside Intel® Core Microarchitecture（Nehalem）》。IEEE[21]Carlson, T.E.、Heirman, W.、Eyerman, S.、Hur, I.、Eeckhout, L.：高层次机制性核心模型评估。《ACM 架构与代码优化汇刊》（TACO）（2014 年）。https://doi.org/10.1145/2629677[22] Bienia, C. 等：PARSEC 基准测试套件：特征分析与体系结构影响

[23]Woo, S.C. 等：《SPLASH-2 程序：特性分析与方法论考量》

[24] Li, S., Ahn, J.H., Strong, R.D., Brockman, J.B.,Tullsen,D.M., Jouppi, N.P.：《面向多核和众核体系结构的 McPAT 框架》。《ACM 架构与代码优化汇刊》**10**（1），1–29（2013年）。https://doi.org/10.1145/2445572.2445577