魏乐源. 基于最近时间的重引用间隔预测策略[J]. 智能计算机与应用,2026,16(1):142-149. DOI:10.20169/j.issn.2095-2163.24032602

基于最近时间的重引用间隔预测策略

魏乐源

(南京邮电大学 计算机学院、软件学院、网络空间安全学院, 南京 210023)

摘 要: 随着计算机技术的飞速发展,处理器与内存之间的性能差距日益凸显。缓存作为计算机体系结构中的关键组成部分,其性能优化对于提升整体系统性能至关重要。替换策略作为缓存管理的核心,对于提高缓存命中率和减少访问延迟具有重要意义。本文提出了一种新颖的缓存替换策略-基于最近时间的重引用间隔预测(LT-RRIP)策略。该策略通过结合最近时间前置过滤器的概念,对传统的重引用间隔预测(RRIP)模型进行改进,旨在降低缓存未命中次数并缩短未命中延迟。通过 SPEC CPU2006 基准测试集的实验验证,LT-RRIP 策略相较于 RRIP 和 LRU 策略,其性能表现均得到了显著提升。

关键词: 缓存内存; 访问时间; 替换策略; 未命中延迟; 最近时间前置过滤器; 重新引用间隔预测

中图分类号: TP333 **文献标志码**: A **文章编号**: 2095-2163(2026)01-0142-08

Latest Time-based Re-Reference Interval Prediction strategy

WEI Leyuan

(School of Computer Science, School of Software, School of Cyberspace Security, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: With the rapid development of computer technology, the performance gap between processors and memory has become increasingly prominent. Cache, as a crucial component in computer architecture, is essential for optimizing overall system performance. Replacement strategies, as the core of cache management, are significant in improving cache hit rates and reducing access latency. This paper proposes a novel cache replacement strategy, namely the Latest Time-based Re-Reference Interval Prediction (LT-RRIP) strategy. By integrating the concept of the latest time prefilter, the strategy enhances the traditional Re-Reference Interval Prediction (RRIP) model, aiming to reduce cache misses and shorten miss latencies. Experimental verification using the SPEC CPU2006 benchmark test set demonstrates that the LT-RRIP strategy significantly outperforms both the RRIP and LRU strategies in terms of performance.

Key words: cache memory; access time; replacement policy; miss latency; Latest Time Prefilter; Re-Reference Interval Prediction

0 引 言

计算机系统同时运行多个应用程序时需要快速访问数据。由于系统性能取决于检索数据所需的访问时间,所以缓存在内存层次结构中起着重要作用。缓存块位于处理器和系统主内存之间,用于存储主内存经常使用的数据。分析观察到程序执行中大部分时间都用于执行重复的指令,如简单循环或嵌套循环。已知大多数这些指令都位于程序的局部区域,并且反复执行,而其余的指令则很少被访问。所以缓存内存块的工作基于计算机系统的“引用局部性”,引用局部性包括空间局部性和时间局部性。当处理器接收数据块请求或指令时,最可能立即请求下一个位置中的另一个数据块或指令,这被称为空间局部性。而时间局部性是指同一数据块或指令被多次请求。

缓存替换算法决定了进入缓存内存的数据块是否应该替换掉已存在于缓存内存中的数据块。当被淘汰的数据块在最长时间内都没有被请求时,这些算法表现最为高效。这些算法通常通过 2 个重要的性能参数来进行分析:缓存命中率和缓存未命中率。缓存命中发生在请求的数据块在缓存内存中存在时。如果不存在,则被称为缓存未命中。

当前常用的替换策略包括最近最少使用

作者简介: 魏乐源(1998—),男,硕士研究生,主要研究方向:缓存优化算法。Email:1030501380@qq.com。

收稿日期: 2024-03-26

(LRU)、最不经常使用(LFU)、最近最常使用(MRU)、随机替换(RR)[1]和先进先出(FIFO)[2]。其中,LRU 策略会淘汰在缓存内存中最久未使用的数据块。当较老的数据块被访问的概率较高时,通常会使用 MRU,最近访问的数据块会被淘汰。在 LFU 策略中,最不经常使用的数据块会从缓存内存中移除。在 RR 策略中,缓存未命中时会随机替换数据块,并为新进入的数据块腾出空间。在 FIFO 策略中,当缓存内存中没有可用空间时,无论其访问情况如何,将淘汰第一个进入缓存的数据块。基于此,本文提出了一种新颖的缓存替换策略:基于最近时间的重引用间隔预测策略(LT-RRIP)。该策略通过结合最近时间前置过滤器的概念,对传统的重引用间隔预测 RRIP 模型进行改进,旨在降低缓存未命中次数并缩短未命中延迟。

1　相关工作

无论是基于软件、还是硬件的各种技术,现已研发提出了广泛的启发式方法。Belady′s MIN[3]是一种最优的缓存替换策略,旨在最小化缓存未命中的数量。这个策略的核心思想是在每次需要替换缓存中的数据时,选择替换那个将在最长时间内不再被使用的数据块。具体来说,Belady′s MIN 策略会预测未来的访问模式,并选择那些在最远的未来才会被重新引用的数据块进行替换,以最大程度地减少未命中的次数。

数据过滤缓存(Filter Data Cache)[4]是一种设计用于降低 L1 指令缓存(L1 I-cache)功耗的小型缓存。通常大小为 1~2 KB。当数据在过滤缓存中命中时,因其靠近处理器,可以将数据直接发送到处理器,从而降低延迟和能耗,不仅节能、而且高效。当预测到过滤缓存会发生未命中时,可以将其完全跳过,直接访问 L1 缓存。数据过滤缓存的实现可以有效地识别平均 92.2%的未命中情况[5]。在这些情况下,可以跳过数据过滤缓存,直接访问 L1 数据缓存。但是,由于数据过滤缓存的空间较小,用其则会增加缓存未命中率。

在多核处理器架构的平台上,最后一级缓存(LLC)是所有不同核心共享的,以增强性能。然而,这可能导致缓存污染,因为低时间局部性的数据可能会导致高时间局部性的其他数据被驱逐出缓存。基于页面可重用性的缓存划分(PRCP)[6]将所有页面迁移到低可重用性或高可重用性区域,从而更有效地划分了缓存。“低可重用性组”通过页面着色获得 LLC 的最小份额,从而使 PRCP 实现了与最佳的基于进程的静态缓存划分方案相当的性能改进。主要优点在于 PRCP 方法不需要额外的硬件支持,也不需要更改应用程序或预先了解工作负载特征,因此大大降低了开销。

对缓存内存进行分区会导致应用程序根据其需求获得动态容量,从而影响底层的替换策略。为此,研究者提出了一种自适应替换策略未命中曲线估计(RME)[7]方法,用于根据选择的替换策略,估算给定应用程序的动态工作负载模式。通过缓存分区技术,最后一级缓存(LLC)由所有应用程序共享。各应用程序预期完全占用其分配的区域,从而降低跨应用程序干扰和应用程序之间容量不平衡的问题。要有效实施分区技术,需要准确预测各应用程序在共享的最后一级缓存中的性能。这是通过未命中曲线来实现的,未命中曲线是一个离散图形,用于比较缓存未命中次数与缓存大小。

这种分区方案可以与现有的强大替换策略有效配置,从而提高性能。对于 VLIW(Very Long Instruction Word)处理器,当在不同性能模式之间切换时,必须共享缓存块。在许多架构中,核心之间不断切换任务会导致缓存性能下降,这是由于不断触发缓存调整操作导致的。缓存缩小可能导致性能下降,最坏情况是所有现存数据都被驱逐,导致冷启动。为了克服这个问题,建议使用一种自适应缓存方法[8],该方法跟踪处理器中模式的变化,并减轻了缩小操作期间缓存未命中率急剧增加的影响。处理器模式更改被视为重新配置数据缓存的触发器。在执行缩小操作之前,实施了一个过渡期。在此期间,活动数据被迁移,因此在缓存大小之间实现了平滑过渡。

最近最少使用(LRU)缓存替换策略的实现在可编程门阵列(FPGA)[9]上进行,使用 VHDL 进行编程配置。尽管大多数实现采用了静态设计用于缓存模型,但已经探索了一种可重构的方法,使用独特的硬件设计,因此允许参数根据应用程序规格由处理器动态设置,包括缓存大小和映射。展示了可重构 LRU 缓存的 2 种不同实现方式,分别是基于树的伪 LRU 和传统 LRU。基于树的伪 LRU 需要一个受害者选择算法,用于在缓存未命中时选择需要替换的块。传统的 LRU 实现使用每行缓存内存中的 8 个十六进制数字,这确定了最大的关联性值。

基于声誉缓存管理策略[10]是一种多级缓存策略,根据多级缓存管理技术做出替换决策,并使用多

步历史提示信息来进行升级和降级的决策。降级提示的存在有助于携带与来自较高级别的块相关的参数,而升级提示有助于携带来自较低级别的块的参数。每个进入缓存的块都初始化为其自己的 K 步提示值(KHV)[11]。升级算法基于最初分配给缓存块的数字 KHV 值。升级基于现有块是否具有比传入块更高的 KHV 值来执行。同样,降级算法基于类似的方法将块降级到较低级别。

LRU 策略旨在适用于具有循环访问模式的工作负载[12]。基于 LIP(LRU 插入策略)的 LRU 策略变种,与重新引用计数[13]结合使用,已经发现可以降低与 L2 缓存相关的未命中率。块的链表也称为 RRIP 链,根据块未来被引用的概率进行排序,位于链表头部的块几乎会立即重新被引用。这意味着在缓存未命中时,移除链表末尾的块将会提高性能。在命中时,将块的重新引用计数与固定的阈值进行比较。如果发现重新引用计数更高,缓存块将被提升到最近使用的位置,否则将被移动到最不经常使用的位置。由于大大减少了 LLC 和主存储器之间的引用,这个策略可以显著提高性能。

基于预期命中次数的策略建立在块的预期命中次数与其重用距离的倒数之间的强相关性[14]上。估计未来命中次数较少的块将被优先淘汰。重新引用间隔预测(RRIP)用于测量块经历的命中次数。每个缓存块都关联有一个重新引用预测值(RRPV),用于受害者选择的依据是块预期将获得的进一步命中次数与 RRPV 之间的差异。

平均逐出时间模型[15]利用重用时间和逐出时间作为缓存未命中估计的参数。当前访问和数据块最近一次访问之间的时间间隔被称为重用时间。块的最后一次访问与其最终被逐出之间的时间间隔被称为逐出时间。这也决定了在最后一次访问后,块预计在缓存中驻留的时间。AET 模型基于缓存未命中概率假设,该假设指出,当前访问的块的重用时间高于平均值时,很可能会导致缓存未命中。这是因为块的引用概率,其重用时间高于平均值的概率等于块不再在缓存中可用的概率。

2 基于最近时间的重引用间隔预测策略

本文提出的方法包括一个最近时间预过滤器(Latest Time Prefilter,LTP)和一个重新引用间隔预测器(Re-Reference Interval Prediction ,RRIP)[16]。模块流程如图 1 所示,其目的是使数据块通过最近时间预过滤器的处理,来增强 RRIP 预测器的准确性。缓存数据块被送入预过滤器,并根据其最近时间进行排名。基于阈值参数(理想情况下是平均最近时间),获得块的子集,再将该子集进一步传递给 RRIP 预测器,RRIP 预测器将根据重新引用预测值(RRPV)来逐出数据块。这个新设计的策略基于 2 个操作阶段被命名为 LT-RRIP。

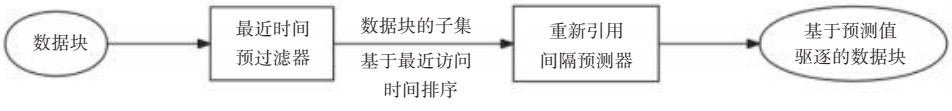

图 1 模块流程

Fig. 1 Process of the module

2.1 最近时间过滤器

在发生缓存未命中时,需要淘汰一个受害缓存块,来为新插入的块腾出空间。淘汰的过程首先根据数据块的最近访问时间进行筛选。最近时间过滤器(LTP)会筛选出一定数量的块,并将这些块的子集交给预测器,以进一步计算要淘汰的块。

每个进入缓存的块都被分配一个最近访问时间。在发生缓存命中或插入新块时,会更新该块在缓存中的最近访问时间。如果发生缓存未命中,会根据所有块的平均最近访问时间设置一个阈值。那些最近访问时间低于设定阈值的块会被传递给预测器进行下一步处理。所以这个机制筛选出最近访问时间较短的块,然后将这些块传递给预测器,以确定哪个块应该被淘汰,从而优化缓存的性能。

2.2 重新引用间隔预测器

从预过滤器中选择要驱逐的块作为预测器的输入。重新引用间隔预测器(RRIP)将每个进入的块与一个重新引用预测值关联起来,并为所有块分配了 2^k 个可能的 RRPV 值,其中 k 为最初时的定义。新进入的块被假定具有长重新引用间隔、即具有较远间隔概率的中间间隔,因此被分配一个 RRPV 值为 $2^k - 2$。在发生缓存未命中时,预测器将驱逐最左边、即第一个 RRPV 值为 $2^k - 1$ 的块、也就是重新引用间隔被预测为较远间隔的块。如果没有任何选择的块的 RRPV 值为 $2^k - 1$,那么预测器将增加所有选择的块的 RRPV 值,直到找到具有此值的块。

在缓存命中的情况下，访问的块的 RRPV 值会减小。

2.3 算法

替换策略在缓存未命中时触发，即引用的数据块在当前缓存存储器中未命中。此时替换策略的作用是找到一个在将来最不可能被引用的块。LT-RRIP 替换策略通过将每个缓存块与一个称为“最近时间”的附加参数关联来实现这一点。在发生缓存未命中时，访问了缓存的所有块，只有那些最近时间小于设置的阈值（所有最近时间的平均值）的块才会传递给重新引用间隔预测器。因此要驱逐的候选块是从这个减少的数据块子集中选择的。相关代码具体如下。

算法 1　LT-RRIP 策略的算法

输入　Data block to be accessed

输出　Data block to be evicted

// All blocks are assigned one of $2k$ possible RRPVs, with k being initially defined

1. function ReplacementPolicy
2. if (cache miss)
   //Recency time prefilter
3. for (all the data blocks in the cache)
4. if (recency time <= set threshold)
5. Pass the data blocks to the predictor for eviction
6. end for
   // RRIP predictor
   // Input: Selected data blocks from the prefilter
7. while (RRPV (of any the selected data block) ! = $2k - 1$) do
8. for (all the selected data blocks)
9. RRPV (data block) ++
10. end for
11. end while
12. evict (leftmost data block with RRPV of $2k - 1$)
13. Add the accessed data block to the cache
14. Associate accessed data block with recency time
15. RRPV (accessed data block) $2k - 2$
16. else // In case of a hit
17. Update the recency time of the accessed data block
18. RRPV (accessed data block) --
19. end function

RRIP 预测器要求重新引用预测值为 $2^k - 1$ 的块才能执行驱逐操作。如果没有这样的候选块可用，那么与所有块相关联的 RRPV 值将不断增加，直到至少有一个块满足条件。最左边重新引用预测值为 $2^k - 1$ 的候选块被驱逐，然后新到达的数据块被放入缓存中。接着计算新块的最近时间，同时更新现有数据块的最近时间。对于访问的数据块，按照正常的重新引用间隔预测模型进行操作，其重新引用预测值被分配为 $2^k - 2$。在缓存命中时，与每个数据块相关联的最近时间会更新，访问缓存块的重新引用预测值会按照正常的重新引用间隔预测模型操作减少。

3 实验设计与分析

3.1 实验基准与设计

为了分析使用本文所提出的方法来提高缓存命中率的预期效果，使用 GEM5 模拟器[17]。GEM5 是一个用于计算机体系结构研究的开源模块化系统级模拟器。模块化的方法有助于模拟计算机系统内的特定组件。特点优势之一是其对包括 ARM、MIPS、RISC-V、ALPHA 和 x86 等多种指令集架构（ISAs）的灵活支持。本次模拟选择了 ALPHA 架构。

Standard Performance Evaluation Corporation (SPEC) 提供了全球公认的、标准化的基准测试，用于对许多计算机性能参数进行分析测量。SPEC CPU2006 基准套件长期以来一直被用于 CPU 密集型分析的行业标准基准测试，包括对处理器和其内存子系统的基准测试。在评估阶段使用的来自这个套件的具体基准测试为 NAMD[18]。

在 ALPHA 处理器架构上执行了一系列模拟实验，旨在评估 L1 指令缓存（L1-I）的性能指标，特别是未命中率和未命中延迟，这些性能指标随着缓存大小和关联度的变化而变化。在实验配置中，L1 数据缓存（L1-D）被设定为具有固定参数，即 32 KB 容量和 4 路组关联，而二级缓存（L2 cache）则被配置为 256 KB 容量和 8 路组关联。

3.2 实验结果分析

在实验过程中，通过调整 L1-I 缓存的大小和关联度来观察其对缓存未命中和延迟的影响，同时保持其他缓存级别（L1-D 和 L2）的参数不变作为控制变量。通过这种方式，能够量化 L1-I 缓存参数变化对处理器性能的具体影响，并为缓存层次结构的

优化设计提供数据支持。

未命中率指的是处理器请求的数据及指令在 L1-I 缓存中未找到的频率,而未命中延迟则是指当发生未命中时,处理器必须等待从更高级别缓存或主存储器中检索数据及指令所产生的额外时间延迟。这些性能指标对于理解处理器的运行效率以及指导缓存系统的设计至关重要。实验结果如图 2、图 3 所示。

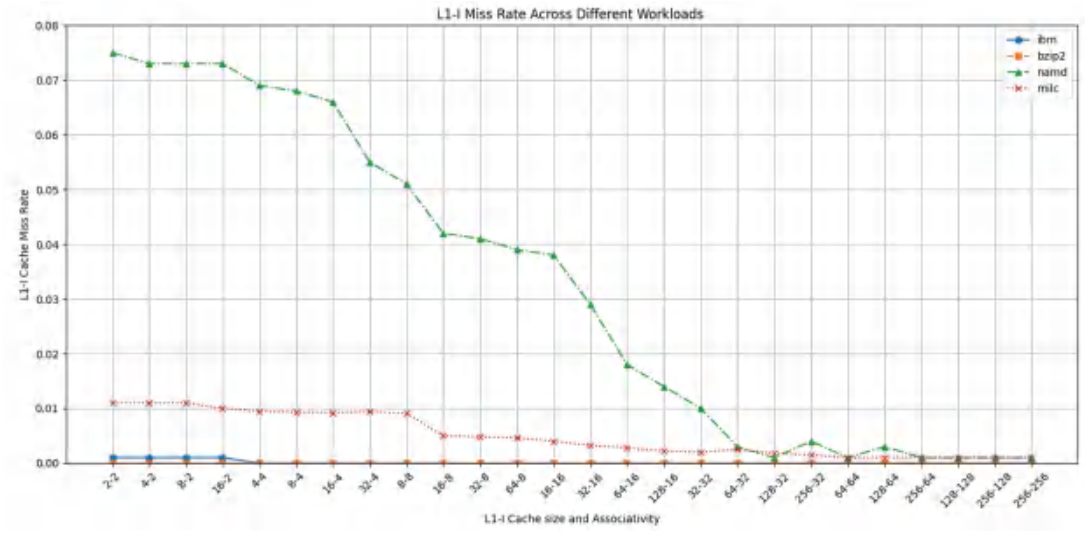

图 2 LT-RRIP 在不同负载下的 L1-I 缓存未命中率变化

Fig. 2 Changes in L1-I cache miss rate of LT-RRIP under different loads

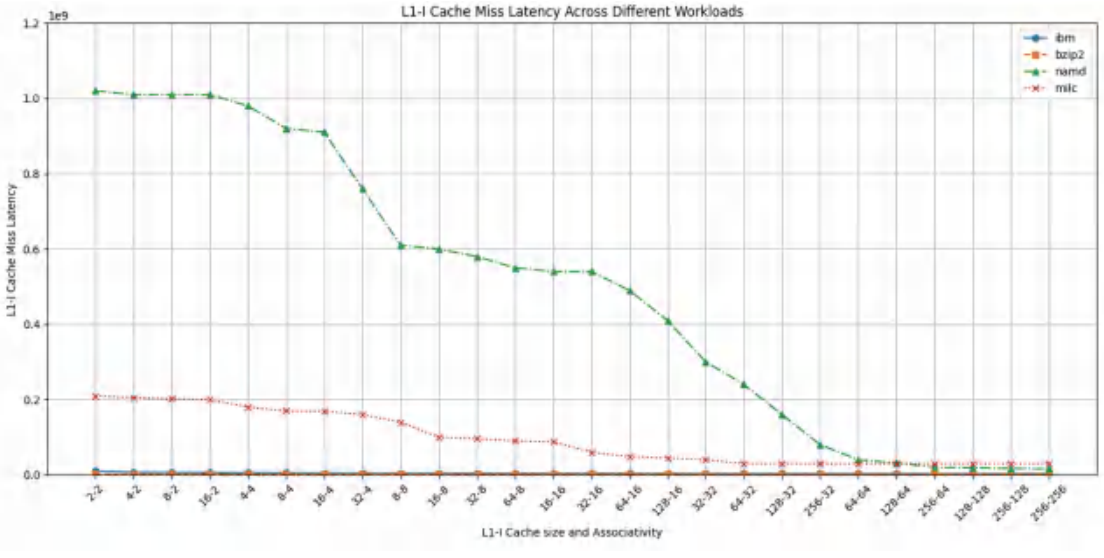

图 3 LT-RRIP 在不同负载下的 L1-I 未命中延迟变化

Fig. 3 Changes in L1-I miss latency of LT-RRIP under different loads

在 L1-D 缓存的一系列缓存大小和关联度值下,对未命中率和未命中延迟的变化进行了分析。其中,L1-I 缓存配置为 32 KB 4 路关联,L2 缓存配置为 256 KB 8 路关联,如图 4、图 5 所示。

在上述情况下,对于某些工作负载,无论缓存大小和关联度如何变化,缓存未命中率和未命中延迟都呈现出下降趋势。而对于其他工作负载,这两者几乎保持在恒定水平,差异微不足道。

对于所有测试案例,当特定工作负载下的缓存大小和关联度增加时,观察到 L1-I 缓存的未命中率和未命中延迟均呈现下降趋势。然而,对于其他工作负载,这些性能指标基本保持不变,显示出微小的差异,可以认为是在一个稳定的水平范围内波动。这表明缓存参数的变化对不同工作负载的性能影响具有差异性,为缓存优化提供了重要的参考依据。

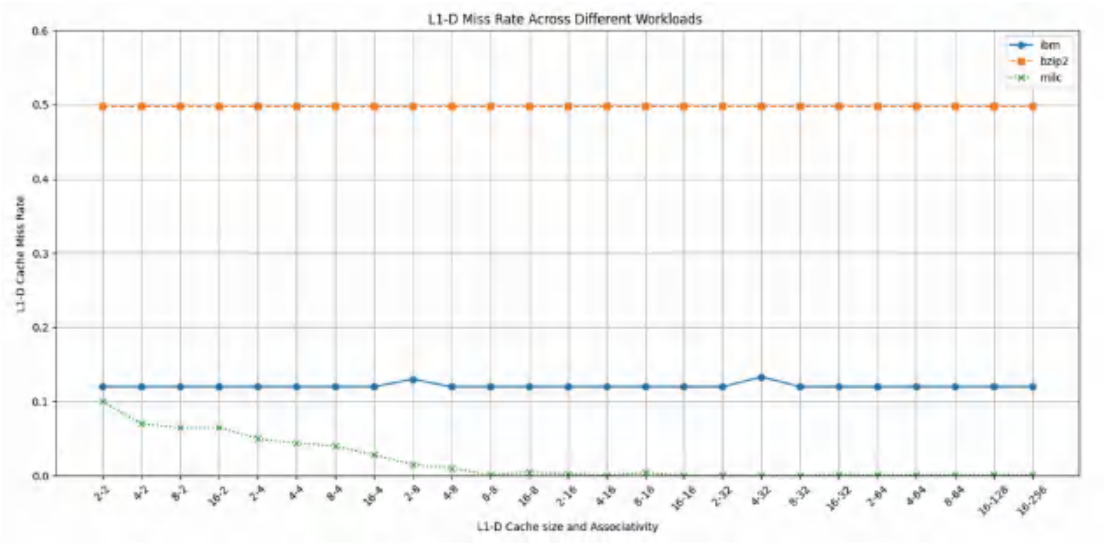

图 4　LT-RRIP 在不同负载下的 L1-D 缓存未命中率变化

Fig. 4　Changes in L1-D cache miss rate of LT-RRIP under different loads

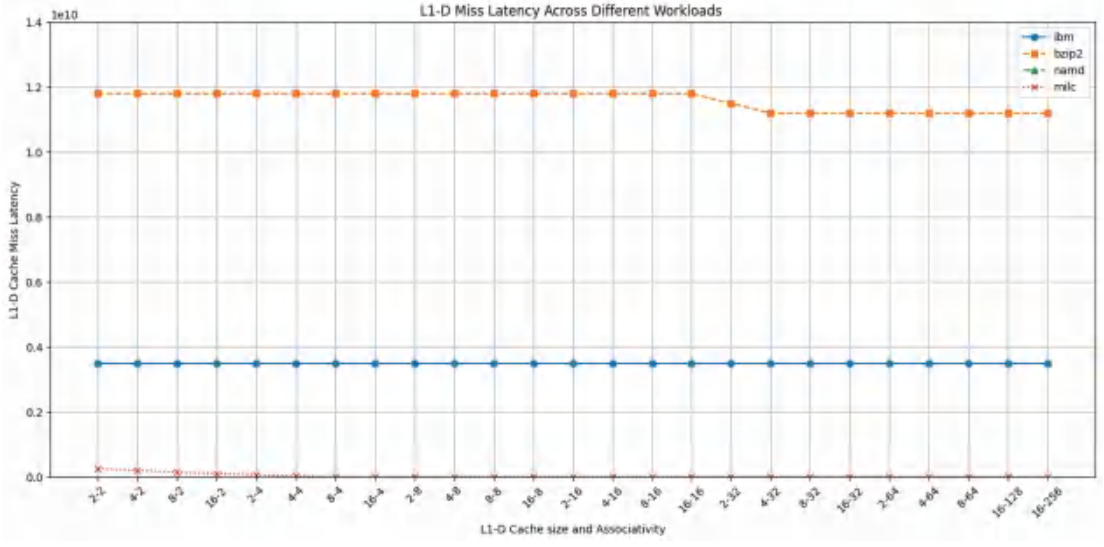

图 5　LT-RRIP 在不同负载下的 L1-D 未命中延迟变化

Fig. 5　Changes in L1-D miss latency of LT-RRIP under different loads

3.3　将 LT-RRIP 与相关工作中提到的策略进行比较

本次研究对新开发的 LT-RRIP 替换策略与现有替换策略(包括 LRU 和 RRIP)进行了详尽的性能对比分析。研究中针对多样化的工作负载进行了全面的性能测试,以确保结果的全面性和准确性。性能评估的核心指标包括平均缓存未命中率、平均未命中延迟以及整体性能提升百分比等。表 1 详细记录了在不同工作负载下,LT-RRIP 相较于 LRU 和 RRIP 策略在平均性能方面的显著提升,为缓存替换策略的选择和优化提供了重要的数据支持。

图 6 展示了在 NAMD 基准和不同 L1-I 缓存大小和关联度值下,LT-RRIP 策略与 RRIP、LRU、LFU 策略的性能比较。

综合上述实验可以发现 LT-RRIP 策略相较于 RRIP 和 LRU 策略,在未命中率与未命中延迟上均有显著的提升。

表 1　LT-RRIP 对比 RRIP 和 LRU 的性能改进

Table 1　Performance improvements of LT-RRIP compared to RRIP and LRU　%

参数	对比 RRIP 缓存策略的提升	对比 LRU 缓存策略的提升
L1-I 未命中率	6.18	2.15
L1-I 未命中延迟	1.94	1.66
L1-D 未命中率	2.92	1.14
L1-D 未命中延迟	1.05	0.78

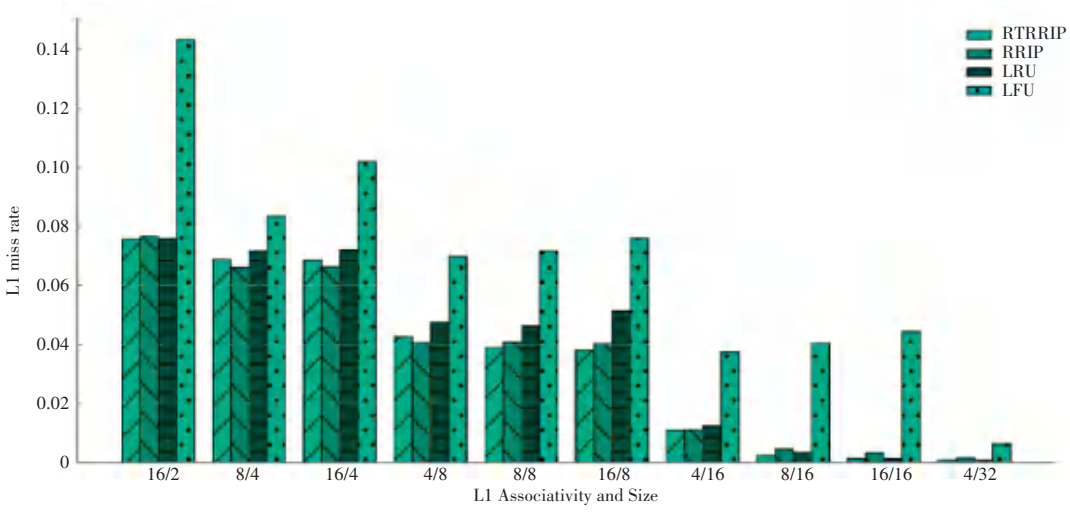

图 6 NAMD 不同工作负载下的未命中率

Fig. 6 Miss rate of NAMD under different workloads

4 结束语

为了突破现有策略的局限性并实现性能提升，研究采用了一种改进的缓存替换策略。该策略以广泛应用的 RRIP 模型为基础，通过引入“最近时间”概念来减少缓存未命中事件并降低未命中延迟。研究中在不同缓存大小和关联度配置下进行了模拟实验，并使用标准化的 SPEC 基准测试集来评估性能。

模拟结果显示，在所有测试的工作负载中，L1-I 缓存的性能得到了显著提升。同时，L1-D 缓存也在多个配置组合中表现出 LT-RRIP 策略相较于其他策略在降低缓存未命中率和未命中延迟方面的优越性。然而，对于 L2 缓存的替换策略，仍有待进一步优化，因为当前策略在多数情况下并未带来明显性能提升。

此外，还基于 LT-RRIP 构建了一种自适应策略，该策略能够根据工作负载的频率特性进行动态调整。由于观察到大多数工作负载在 L2 缓存访问上呈现出频率偏向性，因此这种自适应策略将成为一个额外的性能优势点。

参考文献

[1] KUMAR S, SINGH P K. An overview of modern cache memory and performance analysis of replacement policies [C]// Proceedings of 2016 IEEE International Conference on Engineering and Technology (ICETECH). Piscataway, NJ: IEEE, 2016: 210-214.

[2] KONOVALOV M G, RAZUMCHIK RV. New policy for routing jobs to unobservable parallel queues with FIFO scheduling[J]. Journal of Communications Technology and Electronics, 2021, 66(6): 737-744.

[3] AKANKSHA J, CALVIN L. Rethinking belady´s algorithm to accommodate prefetching [C]//Proceedings of the 45th International Symposium on Computer Architecture (ISCA). Piscataway, NJ: IEEE, 2018: 110-123.

[4] MONTAZERI S S, MOSTAFA R, MA R T B. Frozen cache: Mitigating filter effect and redundancy for network of caches[J]. IEEE Access, 2021, 9: 139725-139738.

[5] XI Xianchang, HUANG Zhikai, NING Lingyi, et al. Real-time pedestrian detection algorithm based on improved YOLOv3[J]. Journal of Physics: Conference Series, 2021, 2002: 012075.

[6] HAYWOOD S R. Self-regulatory organizations; MIAX PEARL, LLC; notice of withdrawal of proposed rule change to amend the MIAX PEARL options fee schedule to remove a monthly credit associated with trading permit fees[J]. The Federal Register / FIND; Lanham, 2022, 87(244): 1.

[7] FAN Zirui, TANG Chen, SHEN Yuxin, et al. RME: A low-light image enhancement model based on reflectance map enhancing[J]. Signal, Image and Video Processing, 2022, 17(4): 1493-1502.

[8] FARHAN B, ANDERS M, BJÖRN L. Effects on nasal airflow and resistance using two different RME appliances: a randomized controlled trial[J]. European Journal of Orthodontics, 2018, 40(3): 281-284.

[9] WEI W, XIN X. Lightweight CNN-based low-light-image enhancement system on FPGA Platform[J]. Neural Processing Letters, 2023, 55(6): 8023-8039.

[10] NESRINE A, AMMARI A C, AHMED A I M, et al. Accelerated and optimized covariance descriptor for pedestrian detection in self-driving cars[J]. Design Automation for Embedded Systems, 2023, 27(3): 139-163.

[11] REICHERT M, MATRAS M, ŁUKASIK A, et al. Host micro-RNA expression in the course of KHV infection in carp[J]. Journal of Comparative Pathology, 2018, 158: 145.

[12] JALEEL A, THEOBALD B K, STEELY C S, et al. High performance cache replacement using re-reference interval prediction (RRIP)[J]. ACM SIGARCH Computer Architecture News, 2010, 38(3): 60-71.

[13] PENG Manman, WANG Peng. Cache replacement using re-

reference interval prediction with adaptive bypassing (RRIPAB) [EB/OL]. (2017-03-01). https://dpi-journals.com/index.php/dtcse/article/view/5664.

[14] HATIBARUAH R, NATH K V, HAZARIKA D. Face image retrieval via sum and difference histograms of elliptical local ternary pattern[J]. CSI Transactions on ICT, 2020, 8(2): 285-291.

[15] RAGHAVENDRA J R, KUNTE S R. Extended local ternary co-relation pattern: A novel feature descriptor for face Anti-spoofing [J]. Journal of Information Security and Applications, 2020, 52: 102482.

[16] YASSIN H Y, JAHRE M, KJELDSBERG G P, et al. Fast and accurate edge computing energy modeling and DVFS implementation in GEM5 using system call emulation mode[J]. Journal of Signal Processing Systems, 2020, 93(1): 33-48.

[17] CEBRIAN M J, BARREDO A, CAMINAL H, et al. Semi-automatic validation of cycle-accurate simulation infrastructures: The case for gem5 - x86 [J]. Future Generation Computer Systems, 2020, 112: 832-847.

[18] HARDYD J H, JAEMIN C, WEI Jiang, et al. Experiences porting NAMD to the data parallel C++ programming model[C]// Proceedings of the International Workshop on OpenCL (IWOCL'22). New York: ACM, 2022: 15.