# 使用重新参考间隔预测的高性能缓存替换

## 概述：

实际的高速缓存替换策略试图通过预测高速缓存块的重新参考间隔来模拟最佳替换。通常使用的LRU替换策略总是预测高速缓存命中和未命中的近似立即参考间隔。表现出远程重参考间隔的应用执行严重下LRU。这样的应用程序通常具有大于缓存的工作集或具有对非时间数据的引用的频繁突发（称为扫描）。为了提高这样的工作负载的性能，本文提出使用重排间隔预测（RRIP）的高速缓存替换。我们建议抗扫描的静态RRIP（SRRIP）和抗扫描和抗篡改的动态RRIP（DRRIP）。两种RRIP策略每个高速缓存块仅需要2位，并且容易地集成到现代处理器中发现的现有LRU近似中。我们在具有2MB最后一级缓存（LLC）的单核处理器上使用PC游戏，多媒体，服务器和SPEC CPU2006工作负载的评估显示，SRRIP和DRRIP在吞吐量指标上的表现优于LRU替换平均4％和10 ％ 分别。

在具有8MB共享LLC的4核CMP上的1000多个多编程工作负载的评估显示，SRRIP和DRRIP在吞吐量度量上分别优于LRU替换平均7％和9％。我们还表明，RRIP的性能优于LFU，这是当前最先进的抗扫描替换算法。对于正在研究的缓存配置，RRIP比LRU需要少2倍的硬件和比LFU少2.5倍的硬件。

## 简介：

最优替换策略使用每个高速缓存块的重新引用（或重用）模式的完全知识来做出其替换决策，并且替换将来将被重新引用最远的块。 另一方面，实际的高速缓存替换策略可以被视为基于它们的替换决策基于哪个块将在未来被重新引用最远的预测并且选择该块用于替换。 通常，在未命中时，替换策略对接下来将重新引用丢失块的时间进行预测。 当关于块的进一步信息可用时，例如在重新引用时，可以更新这些预测。

通常使用的最近最少使用（LRU）替换策略中，LRU链表示用表示最近使用的高速缓存块的MRU位置引用的高速缓存块的新近度，而代表最近最少使用的高速缓存块的LRU位置。最近关于高速缓存插入策略[11,25,28,30]和命中促进策略[30]的建议改变了LRU链的描述。不是表示新近度，而是可以将LRU链视为表示块被预测重新引用的顺序的重参考间隔预测（RRIP）链。预测RRIP链的头部处的块具有接近立即重新引用间隔，而RRIP链的尾部处的块被预测为具有远处的重新参考间隔。近即刻重新引用间隔意味着将在某个时间重新引用高速缓存块，而远程重参考间隔意味着将在较远的未来重新引用高速缓存块。在高速缓存未命中时，将替换RRIP链的尾部处的块（即，预测为将来参考最远的块）。

使用RRIP框架，LRU替换预测填充到高速缓存中的块具有近即刻重新引用间隔，并且因此将其放置在RRIP链的头部。 在重新参考块时，LRU更新其预测，并再次预期块具有近即时重新参考间隔。 实际上，LRU预测高速缓存块以参考的相反顺序被重新引用，即LRU预测最近使用（MRU）高速缓存块将比LRU高速缓存块更快地被重新引用。

虽然LRU为具有高数据局部性的工作负载提供了良好的性能，但是当近似重新引用间隔的预测不正确时，LRU限制性能。 其再引用仅在远端发生的应用在LRU下执行得不好。这种应用对应于应用工作集大于可用高速缓存的情况，或者当对非时间数据的引用突发从高速缓存丢弃活动工作集时。 在这两种情况下，LRU低效地利用高速缓存，因为新插入的块在插入之后没有时间局部性。

动态插入策略（DIP）[25]通过动态地将重参考预测从近即刻重参考间隔改变到远距重参考间隔来改进在重新参考间隔处于远处的情况下的LRU替换 。 在高速缓存未命中时，DIP的LRU插入策略（LIP）组件预测已经驻留在高速缓存中的高速缓存块将比缺少的高速缓存块更快地被重新引用。 结果，当工作集大于可用高速缓存时，LIP通过替换最近填充的高速缓存块而不是使用LRU替换来保留高速缓存中的工作集的一部分。 DIP为工作集大于可用缓存的工作负载动态使用LIP，对所有其他工作负载依赖于LRU。

不幸的是，DIP对工作负载的所有引用进行相同的预测。 DIP的LRU组件预测所有对缺失高速缓存块的重新引用将接近立即，并将它们插入到RRIP链的头部。另一方面，DIP的LIP组件预测所有重新引用丢失的缓存块将在不远的将来，并将它们插入RRIP链的尾部。因此，当工作负载重参考模式混合时，即，近即时和远距重新参考都发生时，LRU和DIP都不能做出准确的预测。例如，当scan [5]从缓存中丢弃应用程序的频繁引用的工作集时，DIP和LRU都会限制缓存性能。扫描被定义为对重新引用间隔在遥远的未来的数据的引用的突发。相比之下，不属于扫描的访问具有接近立即的重新参考间隔。我们的研究表明，许多现实世界的应用程序遭受频繁扫描其缓存访问模式。因此，提高现实世界应用程序的性能需要一个实用的抗扫描缓存替换策略。

扫描，不管它们的长度，在它们的初始引用之后不接收高速缓存命中。这是因为扫描块的重新参考间隔在较远的将来。当扫描第一次开始时，对于驻留在高速缓存中的块，情况可能不同。当在扫描之后引用的数据与在扫描之前引用的数据不同时，扫描期间的替换决定是不相关的，因为对新数据的引用导致强制缺失2。然而，当在扫描之后参考的数据在扫描之前属于工作集合时，最佳替换策略知道远程重新参考间隔被应用于属于扫描的高速缓存块，并且近即刻重新参考间隔是应用于属于工作集的高速缓存块。这样做，最佳替换策略在扫描完成后保留高速缓存中的频繁引用的工作集。实际的替换策略可以通过在扫描期间使用LIPstyle替换以及在没有扫描的情况下使用LRU替换来潜在地实现这一点。

与先前的抗扫描替换算法相比，本文集中于设计高性能抗扫描替换策略，其需要低硬件开销，保留现有的高速缓存结构，并且最重要的是容易地集成到LRU的现有硬件近似中[1,2 ]。 为此，我们提出使用重参考间隔预测（RRIP）的实际替换策略。

RRIP阻止具有远离重参考间隔（即，扫描块）的高速缓存块避免具有近即时重参考间隔（即，非扫描块）的块。 RRIP通过要求每个高速缓存块的M位寄存器来存储其重新预测值（RRPV）来实现这一点。 我们建议抗扫描的静态RRIP（SRRIP）和抗扫描和抗篡改的动态RRIP（DRRIP）。 SRRIP和DRRIP都提高了LRU的性能，并且容易集成到LRU的现有硬件近似中。 事实上，当M = 1时，SRRIP退化为现代高性能处理器中通常使用的不最近使用（NRU）[2]替换策略[1,2]。

本文的其余部分组织如下，第2节激发需要抗扫描替换算法，第3节提供相关工作，第4节介绍RRIP，第5节提供实验方法，第6节介绍结果，最后第7节 总结本文。

有效的最后一级缓存（LLC）利用对于避免长时间缓存未命中到主存储器是至关重要的。 在LRU替换下，许多研究已经表明，通过小缓存过滤时间局部性导致插入到LLC中的大多数块永远不被重新引用[14,16,25,30]。 低效的高速缓存利用率是因为对于由经过滤的时间局部性导致的高速缓存访问模式，LRU执行得不好。

为了更好地理解LRU性能不佳，图1给出了在应用程序中常见的几种典型的缓存访问模式。 令ai表示高速缓存行的地址，（a1，...，ak）表示对k个唯一地址的引用的时间序列，并且令P∈（a1，...，ak）表示以某种概率发生的时间序列 ε。 重复N次的时间序列被表示为（a1，...，ak）N。 缓存访问模式可以分为以下类别：

新近访问模式：图1a显示了一个典型的重复N次的堆栈访问模式。一般来说，新近访问模式具有近即刻重新引用间隔。对于k的任何值，访问模式受益于LRU替换。除LRU之外的任何其他替换策略可以降低这些访问模式的性能。

破坏访问模式：图1b呈现重复N次的长度k的循环访问模式。当k小于或等于高速缓存中的块数时，工作集适合缓存。然而，当k大于高速缓存块的数量时，LRU由于高速缓存抖动而接收零高速缓存命中。对于这样的模式，LRU不提供高速缓存命中，除非增加高速缓存大小以保持存取模式的所有k个条目。当可用高速缓存小于k个条目时，最优替换策略保留高速缓存中的一些工作集。不幸的是，LRU不能这样做。

流存取模式：图1c呈现了流存取模式。当k =∞时，访问模式在其引用中没有局部性。流访问模式可以被表征为具有无限重新引用间隔的工作负载。因此，流存取模式在任何替换策略下不接收高速缓存命中。结果，LRU是足够的，因为在存在流存取模式的情况下替换决定是不相关的。

混合访问模式：混合访问模式可以表征为工作负载，其中一些引用具有近似重新引用间隔，而其他引用具有远程重新引用间隔。图1d示出了使用具有扫描（由图中的灰色框突出显示）的两个示例访问模式。两个示例包括长度为k的访问模式，其重复A次，随后参考具有概率ε的长度m的序列。扫描和访问模式都重复N次。第一参考模式表示对m个条目的链接列表执行操作的应用。初始堆栈参考模式示出了具有到链接列表的开始的时间局部性的操作。扫描示出了需要遍历的搜索或更新操作整个列表

第二示例表示对k个条目的数据结构进行操作并且然后更新m个条目的不同数据结构的应用。对于两种访问模式，当m + k小于可用高速缓存时，总工作集合适合

缓存和LRU工作良好。然而，当m + k大于可用高速缓存时，LRU从高速缓存中丢弃频繁引用的工作集。因此，在扫描之后，对频繁参考的工作集的访问总是错过。在没有扫描的情况下，混合访问模式优选LRU。但是，在存在扫描的情况下，最佳策略在扫描完成后保留高速缓存中的活动工作集。不幸的是，LRU不能保留活动工作集。

LRU用于抖动和混合访问模式。 DIP [25]通过保留高速缓存中的一些工作集来解决高速缓存抖动问题。不幸的是，DIP只针对工作集大于可用缓存的工作负载，并且依赖于所有其他工作负载的LRU。因此，DIP限制了工作负载的性能，其中频繁扫描丢弃来自高速缓存的活动工作。为了说明这个问题，图2比较了抗闪存DIP和耐扫描HYBLRU / LFU的缓存性能。 HYBLRU / LFU是一种混合高速缓存替换策略，其使用集合决策[25]在抗扫描最不频繁使用（LFU）3 [17]替换策略和LRU替换之间动态选择。该研究包括14个工作负载，每个工作负载在2MB LLC上运行。 yaxis表示与LRU相比的高速缓存未命中的平均减少，而x轴表示应用及其类别。所研究的应用类别是PC游戏，多媒体，服务器和SPEC CPU 2006。

图2显示，对于SPEC CPU2006类别的两个工作负载，DIP优于LFU。 然而，LFU在PC游戏和多媒体类别（以及SPEC CPU2006类别中的hmmer工作负载）的工作负载方面显着优于DIP。 例如，PC游戏和多媒体工作负载没有观察到来自抗窜改DIP的好处，但是观察到来自抗扫描LFU的高速缓存未命中的高达20％的减少。 图2中的结果激发了对实际的高速缓存替换策略的需要，该策略不仅是抗篡改的而且是抗扫描的。 第4节讨论这种替代政策的设计。

## 相关工作

工业和学术界已经产生了大量的研究工作，致力于改善替代政策的绩效。虽然我们不能描述文献中存在的所有替换策略，但是我们总结了最密切地涉及通过定向在高速缓存插入时死的高速缓存块来提高LLC性能的现有技术。

通过扫描引入的死块通常通过使用访问频率来预测重新参考模式来解决。所提出的最频繁使用（LFU）替代策略预测频繁访问的块将在近即将来重新引用，而不频繁访问的块将在远的将来被重新引用。 LFU通过使用计数器来测量块的访问频率来实现这一点。虽然LFU通过频繁扫描提高了工作负载的性能，但它显着降低了新近性是替换的首选选择的工作负载的性能。几项研究综合了近期和频率[17,23,27]来解决这个问题，但是它们需要在每个工作负载基础上调整几个参数。存在自调整自适应策略[22,5,29]，然而它们显着增加了硬件开销和复杂性。通过混合高速缓存替换可以降低硬件开销和复杂性[25]。混合缓存替换使用集合决斗[25]在多个替换策略之间动态选择。虽然使用LRU和LFU的混合高速缓存替换可以提供扫描抵抗，但混合替换需要用于两种不同高速缓存替换策略的硬件和验证开销。非常希望单个高速缓存替换策略提供扫描抵抗并且对于新近友好工作负载执行良好。

另一研究领域预测高速缓存块的重新参考间隔何时变得遥远，即高速缓存块变为死[16,18]。 最近的研究应用了死区预测在LLC [19]。 所提出的策略利用每个高速缓存块附加预测以确定该块是否死。 所提出的策略使用块的重参考历史来预测在块移出RRIP链的头部之后的死亡。 受害者选择策略选择更接近RRIP链的尾部的死块。 虽然死块预测改善了高速缓存性能，但它需要用于死块预测器的附加硬件开销。

最近的一项研究[25]表明，当应用程序工作集大于可用缓存时，发生死块。 在这种情况下，所提出的动态插入策略（DIP）动态地改变插入策略从总是插入在RRIP链的头部的块，以插入大部分块在RRIP链的尾部。 通过这样做，DIP保留了缓存中的一些工作集。 由于DIP为工作负载的所有引用做出单个插入策略决定，所以DIP仅针对工作集大于可用高速缓存的工作负载。因此，在存在扫描的情况下，DIP的LRU组件策略不能保持活动工作设置在缓存中。

另一个最近的研究提出了基于伪LIFO [8]的替代策略。 策略建议使用填充栈而不是新近性栈来进行高速缓存替换。 所提出的策略学习超出每个填充堆栈位置的高速缓存块的重新引用概率，并且发现从填充堆栈的上部分驱逐块通过快速地驱逐死块来提高高速缓存利用率。然而，所提出的策略需要额外的硬件来保持跟踪 填充堆栈位置，并且还需要动态机制来学习填充堆栈上的最佳驱逐位置。

存在各种其他解决方案[6,13,20,26,31]，但是它们或者需要大量的附加硬件，或者它们显着地改变现有高速缓存的组织。 重用距离预测[15]最接近地类似于本文提出的工作。 重用距离预测通过使用PC索引预测器明确地计算给定高速缓存块的重用距离。 RRIP没有明确计算复用距离。 相反，RRIP总是预测所有丢失的缓存块将具有相同的重新引用间隔，并且当更多信息可用时（例如，在重新引用时）更新预测。 RRIP也不同于现有技术，因为其提出了不需要显着的硬件开销或对现有高速缓存结构的改变的高性能的实用的抗扫描高速缓存替换策略。

## RRIP

当工作负载具有混合访问模式时，LRU替换及其近似不能完美地区分具有远离重新引用间隔的块与具有近近重新引用间隔的块。 由于基于链的LRU替换对于建立用于高度关联高速缓存的硬件是不切实际的，我们示出了使用不最近使用（NRU）[1,2]替换策略的混合访问模式的问题。

### NRU

不最近使用（NRU）替换策略是在现代高性能处理器中常用的LRU的近似。 NRU使用称为nru-bit4的每个高速缓存块的单个比特。由于只有一个比特的信息，NRU允许两个重新参考间隔预测：近即时重参考和远距离重参考。通信比特值“0”意味着最近使用了块，并且该块被预测为在近期将被重新引用。 nru位值'1'意味着该块最近未被使用，并且该块被预测为在远的未来被重新引用。在缓存填充时，NRU总是预测丢失的块将具有近即时重新引用。在重新引用时，NRU再次预期引用的块将具有近即时重新引用。在高速缓存未命中时，NRU选择其预测的重参考在远的将来的牺牲者高速缓存块，即，其nru位为“1”的块。由于多个块可能具有远处的重参考预测，因此需要仲裁器。 NRU受害者选择策略总是从固定位置（在我们的研究中的左侧）开始受害者搜索。在所有nru比特为'0'的情况下，即，所有块被预测为在近即将的将来被重新引用，NRU更新所有高速缓存块的重参考预测以处于遥远的未来，并重复受害者搜索。将所有nru位更新为“1”允许受害者选择策略提高进度，同时从高速缓存中删除过时的块。

图3b示出了使用用无效块'I'初始化的4条目高速缓存的NRU的行为。 作为参考，图3a还示出了LRU的行为。 该图还示出了替换策略对高速缓存命中和高速缓存未命中采取的步骤。 我们使用以下双模式访问模式来说明NRU行为：



图3b示出了高速缓存的四个块，每个块具有在右下角示出的nru位。 该图示出了在扫描完成之后，当最优替换策略将在高速缓存中保留它们时，对a1和a2的引用都丢失高速缓存

### SRRIP

利用仅一个比特的信息，NRU可以为填充到高速缓存中的所有块预测近似重新参考间隔或远距离重新参考间隔。 总是预测所有高速缓存插入上的近即时重新引用间隔限制混合访问模式的高速缓存性能，因为扫描块不必要地占用高速缓存空间而没有接收任何高速缓存命中。 另一方面，总是预测远距离再参考间隔显着地降低主要具有近即时重新参考间隔的接入模式的高速缓存性能。 因此，对于每个丢失的高速缓存块，没有关于重参考间隔的任何外部信息，NRU不能识别和保留混合存取模式中的非扫描块。

为了解决NRU的限制，我们增强了与每个高速缓存块存储的重参考预测的粒度。 我们提议基于重参考间隔预测（RRIP）的高速缓存替换.RRIP使用每个高速缓存块的M位来存储2M个可能的重排预测值（RRPV）中的一个。 RRIP动态地学习高速缓存访问模式中的每个块的重新引用信息。 与NRU一样，RRPV为零意味着高速缓存块被预测为在近近的将来被重新引用，而饱和的RRPV（即，2M-1）意味着高速缓存块被预测为在 遥远的未来。 定量地，RRIP预测具有小RRPV的块比具有大RRPV的块更快地被重新引用。 当M = 1时，RRIP与NRU替换策略相同。 当M> 1时，RRIP启用大于近即刻重新参考间隔但小于远距离重新参考间隔的中间重新参考间隔。

RRIP的主要目标是防止具有较远重新引用间隔的块污染缓存。在没有任何外部重参考信息的情况下，RRIP静态地预测块的重新参考间隔。由于总是预测在高速缓存插入时间处的近即时或远距离重参考间隔在所有访问模式中不鲁棒，所以RRIP总是插入具有长重新参考间隔的新块。长的重参考间隔被定义为偏向远距离的重参考间隔的中间重参考间隔。我们使用2M-2的RRPV来表示长的重参考间隔。总是在缓存插入时预测长的重参考间隔的直觉是防止具有在远的将来的重新引用的缓存块污染缓存。另外，总是预测长的重参考间隔而不是远距离的重参考间隔允许RRIP有更多的时间来学习和改进重参考预测。如果新插入的高速缓存块具有近即时重新参考间隔，则RRIP可以将重新参考预测更新为比先前预测短。实际上，RRIP学习块的重新参考间隔。

在高速缓存未命中时，RRIP受害者选择策略通过找到预测在遥远的未来被重新引用的第一块（即，其RRPV为2M-1的块）来选择受害者块。与NRU相似，受害者选择策略断开 通过始终从固定位置（在我们的研究左侧）开始受害者搜索。 在RRIP不能找到具有远的重新参考间隔的块的情况下，RRIP通过增加高速缓存集合中的所有块的RRPV来更新重新参考预测，并重复搜索，直到具有远处重新参考的块 间隔。 在受害者选择时更新RRPV允许RRIP通过从缓存中移除过时的块来适应应用工作集中的改变。

改变块的重参考预测的自然机会发生在对块的命中。用于RRPV寄存器的此更新的算法称为RRIP命中提升策略。命中提升策略的主要目的是动态地提高高速缓存块的预测重新引用间隔的准确性。我们提出两个策略来更新重新参考预测：命中优先级（HP）和频率优先级（FP）。 RRIP-HP策略预测接收命中的块将在近即将来被重新引用，并将相关块的RRPV更新为零。 HP策略的目标是优先处理在接收命中的任何高速缓存块上不接收高速缓存命中的块的替换。但是，当高速缓存块在缓存插入后仅重新引用一次时，HP策略可能会降低缓存性能。在这种情况下，HP策略不正确地预测接近立即重参考预测而不是块的远距离重参考预测，并且使得该块占用有价值的高速缓存空间而不接收任何命中。为了解决这个问题，RRIP-FP策略使用更多信息（即，高速缓存命中）来更新重参考预测。代替将重新引用预测更新为在命中时接近即时，RRIP-FP将每次块接收到命中时将预测的重参考间隔更新为短于先前的重参考间隔。 FP策略通过在高速缓存命中时减少RRPV寄存器（除非RRPV寄存器已经为零）来实现此目的。 FP策略的目标是优先考虑在频繁重新引用的高速缓存块上替换不频繁重新引用的高速缓存块。

由于由RRIP进行的重参考预测是在高速缓存命中和未命中时静态确定的，所以我们将该替换策略称为静态重参考间隔预测（SRRIP）。 图3c说明了2位SRRIP-HP的行为。 该示例示出了SRRIP通过为主动使用的高速缓存块和扫描块的远距离重参考间隔正确预测近似重新参考间隔来模拟最佳替换。

一般来说，对于关联性A，主动工作集大小w（w <A）和扫描长度Slen，M位SRRIP是抗扫描的

                   Slen <=（2M-1）\*（A-w）（等式1）

当等式1中的条件不成立时，由于受害者选择策略的老化机制，SRRIP无法保留高速缓存中的活动工作集，因为扫描块和非扫描块的RRPV变得相同。 在这种情况下，可以通过增加RRPV寄存器的宽度来保持活动工作集更长的时间。尽管大的RRPV可以抵抗长扫描，但是当高速缓存块接收到其最后命中时，它们可能导致低效的高速缓存利用率， RRPV变为零。 在这种情况下，有效高速缓存容量减少，直到牺牲选择策略更新死块的重参考预测以具有远程重参考。 因此，使用RRIP的扫描电阻要求将RRPV寄存器的宽度适当地调整大小以避免性能降级的来源。

### DRRIP

当所有块的重新引用间隔大于可用高速缓存时，SRRIP无效地利用高速缓存。 在这种情况下，SRRIP会导致缓存抖动，并导致没有缓存命中。 为了避免高速缓存抖动，我们提出了用远距离重参考间隔预测（即，RR为2M-1）插入大多数高速缓存块并且偶尔（具有低概率）插入具有长重复的高速缓存块的双模RRIP（BRRIP） 参考间隔预测（即，2M-2的RRPV）。 BRRIP类似于DIP的双模式插入策略（BIP）[25]组件，它帮助保存高速缓存中的一些工作集。

对于非颠簸访问模式，始终使用BRRIP可以显着降低缓存性能。为了在所有缓存访问模式下都稳健，我们建议动态确定应用程序是否最适合于抗扫描SRRIP或抗阻BRRIP。我们建议使用集合决策[25]的动态重参考间隔预测（DRRIP）来识别哪个替换策略最适合应用程序。DRRIP通过使用两个集合决斗监视器（SDM）[11]动态地选择抗扫描SRRIP和抗冲击BRRIP之间。SDM通过永久地使几个集合5的缓存遵循该策略来估计任何给定策略的未命中。Set Dueling使用单个策略选择（PSEL）计数器来确定获胜策略。DRRIP对高速缓存的其余集合使用两个SDM的获胜策略。

### SRRIP与LRU对比

修改LRU管理的缓存以预测重新引用间隔的自然方式将是通过改变LRU链上的块的插入位置，从而创建RRIP链。 基线MRU插入策略（MIP）[25]通过始终在RRIP链的头部插入新块来预测近即刻重新引用间隔。 LRU插入策略（LIP）[25]通过总是在RRIP链的尾部插入新块来预测远程重新引用间隔。 RRIP链中间的插入位置[28,30]可以用于预测中间重参考间隔。尽管RRIP链中的不同插入位置可以提供扫描电阻，但是它们需要在每个应用的基础上进行调整。 当工作集中的变化需要替代的重参考预测，即RRIP链上的替代插入位置时，所有应用的静态插入位置可能降低性能。

与SRRIP不同，修改的LRU不能自动适应应用程序工作集大小的更改，因此可能显着降低性能。集合决斗可以用于设计动态地标识适合于应用（或应用阶段）的RRIP链上的最佳插入位置的机制。具体地，SDM可以监视RRIP链上的不同插入位置的性能，然后应用最佳位置到缓存的其余部分。然而，使用SDM来标识RRIP链上的最佳插入位置不能随着缓存相关性的增加而扩展。这是因为所需的监视集数量可能超过缓存中的总集数。因此，设置决斗控制策略以识别RRIP链上的最佳插入位置不被认为是提供扫描抗性的实际解决方案。尽管如此，我们将SRRIP和DRRIP与基于每个应用程序知道RRIP链上最佳单插入位置的离线分析机制进行比较。我们将此方案称为最佳离线插入策略（BOIP）。我们不考虑潜在的更好和更复杂的方案，其中RRIP链上的插入位置动态地适应于应用的不同阶段。

1. **RRIP扩展到共享缓存**

随着芯片内核数量的增加，共享高速缓存现在非常普遍。 由于共享高速缓存从并发执行的工作负载接收访问模式，所以来自不同工作负载的组合访问流也可以被认为是混合访问模式。因此，SRRIP自然延伸到共享缓存，可以用不同的内存需求的应用程序之间尽量减少高速缓存争。 例如，使用混合访问模式术语，对活动工作集合的引用可以被其工作集小并且适合于共享LLC的应用潜在地描述为存储器引用，并且扫描可以被应用描述为存储器引用 与非常大的工作集。 在这种情况下，SRRIP通过保留共享LLC中的小工作集来减少缓存争用。

将DRRIP扩展到共享高速缓存类似于扩展DIP到共享高速缓存。 我们提出了类似于线程意识动态插入策略（TA-DIP）的线程意识DRRIP（TADRRIP）[9]。 TA-DRRIP每个应用程序使用两个SDM来动态确定应用程序是否应在其他应用程序存在的情况下使用SRRIP或BRRIP。 与TA-DIP一样，TADRRIP仅需要用于共享LLC的每个硬件线程的策略选择计数器（PSEL）。

1. **实验方法**

### 1、模拟器

我们使用CMP $ im [10]，一个基于Pin [21]的跟踪驱动的x86模拟器，用于我们的性能研究。我们的基线处理器是4路无序的，具有128个条目的重排缓冲器和三级缓存层次结构。只有层次结构的LLC强制包含。我们的缓存层次结构与Intel Core i7 [3]大致相当。 L1指令和数据高速缓存是4路32KB，而L2高速缓存是统一的8路256KB。在我们的研究中，L1和L2高速缓存大小保持不变。我们在数据高速缓存上支持两个L1读端口和一个L1写端口。我们评估单核和4核配置。在4核配置中，L1和L2是私有的，只有LLC由所有四个核共享。基线LLC（L3）在单核中为16路2MB，在4核系统中为8MB。层次结构中的所有高速缓存都使用64B行大小。对于替换决策，层次结构中的所有高速缓存都使用LRU替换策略。只有对高速缓存的请求引用更新LRU状态，而非请求引用（例如，回写引用）使LRU状态保持不变。 L1，L2和L3缓存的负载到使用延迟分别为1,10和24个周期。我们对主内存建模250周期惩罚，并支持最多32个未完成的内存缺失。

### Benchmarks

对于我们的单核研究，我们使用SPEC CPU2006基准套件的五个工作负载和来自PC游戏，多媒体和服务器工作负载段的九个“真实世界”工作负载。 选择这些工作负载是因为它们对基线处理器配置上的存储器延迟敏感，并且有机会通过增强的替换决策来提高其性能。 为了更全面，我们在附录中报告了大量内存密集型和非内存密集型工作负载的结果。 SPEC CPU2006工作负载都是使用PinPoints [24]作为参考输入集收集的，而现实世界的工作负载都是在硬件跟踪平台上收集的。 实际工作负载包括操作系统和用户级活动，而SPEC CPU2006工作负载仅包括用户级活动。 表1列出了所使用的工作负载，图4提供了对不同高速缓存大小的敏感性。 工作负载全部运行250M指令

对于我们的多核工作负载，我们创建了所有可能的4核组合（14选择4 = 1001工作负载）。 运行模拟，直到所有基准运行250万条指令。 每个核心的统计只收集了前250万条指令。 如果到达跟踪结束，模型将回绕跟踪并从头开始重新启动。 仿真方法与最近在共享缓存上的工作类似[11,8,30,20]。

1. **结果与分析**

### SRRIP对插入RRPV的敏感性

图5显示了SRRIP-HP对M位寄存器宽度的灵敏度，以及当两者都静态改变时，缓存插入时的Re-reference Prediction Value。 y轴表示与LRU替换相比的高速缓存未命中的百分比减少。 对于M = 1（NRU），2,3,4和5，x轴示出了用于高速缓存插入的所有2M个可能的RRPV。 x轴标记遵循格式“INS = r，M = m”，并且表示m位SRRIP配置，其中所有丢失的高速缓存块被插入具有'r'的RRPV。 对于每个SRRIP配置，该图还显示了减少所有工作负载的高速缓存未命中的最大值，平均值和最小值。 平均值由正方形表示，而最小值和最大值由三角形表示。

图5示出了当M> 1时，总是预测丢失的高速缓存块具有长的重参考间隔具有最佳性能。 事实上，预测长的重复参考间隔始终优于NRU替换（M = 1）。 这是因为RRIP增强了用于预测重新参考间隔的粒度。 通过总是预测长的重参考间隔，没有时间局部性的高速缓存块（即，扫描块）在长时间段内不会污染高速缓存。 总是预测远距离重参考间隔具有最差的性能，因为SRRIP没有足够的时间来改善块重新参考间隔。 虽然总是预测远距离重参考间隔具有正离群值，但是总是预测长的重参考间隔在所有工作负载上是鲁棒的，并且相对于LRU减少高速缓存未命中6-10％。

### SRRIP性能

图6显示了M = 1,2,3,4和5的SRRIP-HP和SRRIP-FP的每个工作负载行为。两个SRRIP策略总是预测缓存插入时的长重复参考间隔。 x轴显示不同的工作负载，而y轴显示高速缓存未命中的减少。 X轴标签GAMES，MULTIMEDIA，SERVER和SPEC06表示这些类别中工作负载的平均值，而ALL是所有14个工作负载的平均值。我们对缓存性能和系统性能的几何平均值使用算术平均值。图6a显示SRRIP-FP将MPKI降低5-18％。 MPKI的减少允许SRRIP-FP在所有工作负载中平均超过LRU超过2.5％（图6c）。 PC游戏获得最大的优势，其中SRRIP-FP的平均性能优于LRU 4％。另一方面，SRRIP-HP为十四个工作负载中的八个降低了5％-15％的MPKI（图6b）。 MPKI的减少允许SRRIP-HP在所有工作负载中优于LRU 5％（图6d）。 PC游戏和多媒体工作负载比LRU最有利8-11％。这些结果与图2一致，其中PC游戏和多媒体工作负载受益于扫描抗性。

平均来说，当M> 3时，SRRIP对RRPV寄存器的宽度不敏感。 当RRPV寄存器的宽度增加时，某些工作负载的性能会下降。 这是因为更宽的RRPV寄存器在高速缓存中保留过时的块（在它们的最后一次命中之后）并且减少有效的高速缓存容量。 对于本研究中的工作负载，2位或3位RRPV足以耐扫描

最后，SRRIP-HP和SRRIP-FP都胜过LRU。 NRU（M = 1）几乎总是比LRU差。 此外，SRRIP-HP提供了SRRIP-FP两倍的性能优势。 这意味着抗扫描替换算法的第一阶优点不是从高速缓存中精确检测频繁引用的数据，而是从保留接收高速缓存命中的数据，即活动工作集合。 对于本文的其余部分，除非另有说明，我们仅提供SRRIP-HP的结果。

### SRRIP对缓存配置的敏感性

图7显示了不同LLC大小的不同工作负载类别的SRRIP性能：512KB，1MB，2MB，4MB和8MB。 所有LLC都是16路集合关联。 y轴示出了相对于相应LLC的LRU替换的性能。 该图示出对于所有高速缓存大小，NRU（M = 1）总是类似于LRU。 然而，对于各种高速缓存大小，SRRIP的LRU优于5-20％。 我们还通过改变缓存相关性从4路到128路进行了SRRIP灵敏度研究。 我们的研究产生了与图6相当的结果。这些结果表明SRRIP可扩展到不同的高速缓存配置。 由于大多数性能增益是通过3位RRPV寄存器实现的，因此我们仅关注2位和3位SRRIP

### DRRIP性能

图8显示了2位和3位DRRIP6的性能。图8a显示，DRRIP可显着提高SPEC CPU2006工作负载sphinx3，hmmer和mcf的高速缓存性能。这些工作负载在工作集中具有略大于2MB高速缓存的膝盖（参见图4）。 PC游戏和多媒体工作负载也受益于缓存缺失的减少。另一方面，服务器工作负载在工作集中没有拐点，因此没有观察到DRRIP的好处。在大多数工作负载中，DRRIP具有与SRRIP类似或更好的性能。 DRRIP只会伤害photoshop性能，尽管缓存未命中减少10％。进一步的分析表明，photoshop对于经常在扫描之间参考的记忆区域非常敏感。由于DRRIP针对缓存缺失度量而不是吞吐量度量进行优化，因此当应用程序中的缺失成本发生变化时，DRRIP可能会降低性能。增强DRRIP以针对吞吐量而不是缓存缺失进行优化可以解决Photoshop的问题。然而，平均来说，DRRIP的性能比SRRIP提高了5％。由于2位和3位DRRIP的性能类似，我们得出结论，2位DRRIP足以满足扫描电阻和thrash电阻。因此，对于本文的剩余部分我们只关注2位RRIP。

### 共享缓存上的RRIP

图9给出了SRRIP和TADRRIP与吞吐量度量的LRU相比的性能的s曲线。 x轴表示所有1001个多编程工作负载，而y轴表示相对于LRU的性能。 SRRIP在4核CMP上提高性能高达25％，而TA-DRRIP将性能提高了2.1倍。 在研究的所有工作负载中，SRRIP不会降低任何工作负载的性能，而对于少于25个工作负载，由于使用BRRIP SDM进行实验的成本，TA-DRRIP将性能降低2-5％。 平均而言，在1001个多编程工作负载中，SRRIP将性能提高了7％，而TA-DRRIP将性能提高了10％。 因此，这些结果显示SRRIP和TA-DRRIP都是稳健的和高性能的

### RRIP在不同的缓存级别

我们比较了SRRIP到LRU的性能，当应用在我们的三级层次结构的L1和L2缓存时。 在L1高速缓存中，SRRIP不提供改善性能的机会，因为高速缓存大小太小并且时间局部性太高。 在L2高速缓存中，SRRIP没有提供显着的性能提升，因为L2高速缓存很小（在我们的研究中为256KB）。 SRRIP没有降低L1或L2高速缓存的性能。 为了确保SRRIP在LLC上表现良好，我们通过删除L2缓存将我们的层次结构从3级层次修改为2级层次结构。 对于这个2级层次结构，SRRIP和DRRIP的性能分别优于LRU 4.8％和10％。 因此，RRIP最适用于LLC，其中时间局部性由层次结构的较小级别过滤7。

RRIP要求每个高速缓存块有2位寄存器。 RRIP集成到现有的NRU实现中，对受害者选择硬件进行较小的修改。 NRU受害者选择策略搜索nru位值为“1”的第一块。另一方面，SRRIP搜索其再参考间隔在未来最远的第一高速缓存块，即其RRPV在集合中最大的块。可以通过复制查找第一个（FFO）逻辑来实现搜索。对于2位RRIP，需要四个FFO电路（具有适当的输入）来找到指向第一个'0'，'1'，'2'和'3'RPV寄存器的指针。优先级MUX选择适当的FFO电路的输出作为受害者。在没有找到具有远离RRIP的块的情况下，RRIP还需要额外的逻辑来老化RRPV寄存器。 NRU通过简单地反转集合中的所有nru个比特来对高速缓存块进行老化。 SRRIP需要状态机逻辑来对集合中的所有RRPV寄存器进行老化。 DRRIP和TA-DRRIP仅需要每线程10位策略选择（PSEL）计数器和用于选择SDM的逻辑。 SRRIP和DRRIP的设计更改不在关键路径上，因此不影响高速缓存访问时间。

### 将RRIP与其他策略进行比较

对于我们研究中的14个工作负载，表3比较了以下替换策略对LRU替换的性能：NRU，SRRIP，peLIFO [8]，DIP，HYBLRU / LFU，DRRIP和BOIP。 图10呈现了每个应用程序的这些替换策略的性能比较。 DIP使用集合决斗在近即时和远距离重参考间隔预测之间动态选择。 DIP和peLIFO使用NRU替换作为基线替换策略.HYBNRU / LFU还使用集合决策来动态地在NRU和LFU替换之间进行选择。 BOIP使用离线分析信息来确定适合于应用程序的RRIP链上的最佳静态插入位置。 peLIFO跟踪填充堆栈上的缓存命中以指导缓存替换。 SRRIP，非适应策略，胜过LRU替换，而DRRIP的性能优于最佳性能的抗扫描混合高速缓存替换策略（HYBNRU / LFU）分析信息（BOIP）。

我们还将共享缓存上的SRRIP和TA-DRRIP的性能与peLIFO和TA-DIP进行了比较。 peLIFO和TA-DIP都使用NRU作为基线替换策略。 对于我们的研究中的1001多程序工作负载，我们发现TA-DIP相对于LRU提高了4％的性能，而peLIFO相对于LRU提高了1.5％的性能。 另一方面，SRRIP和TADRRIP分别提高了7％和9％的性能。 我们认为我们对peLIFO的结果与[8]不同，因为我们对更广泛的工作负载进行评估。

RRIP需要比LRU替换更少的硬件，但是平均优于LRU替换。对于n路关联高速缓存，LRU替换需要每个高速缓存集合n⋅log2n个比特，而RRIP仅需要每个高速缓存集合2n个比特。 与HYBNRU / LFU相比，LFU组件策略需要用于频率计数器的硬件，并且NRU组件需要用于跟踪新近性的硬件。 假设LFU频率计数器有4位，HYBNRU / LFU每个高速缓存集需要5n位。 此外，混合替换还需要用于设计两种不同替换策略的验证开销。 相比之下，SRRIP和DRRIP在单个替换策略中提供了扫描抗性和抗击穿性。 RRIP比HYBNRU / LFU需要少2.5倍的硬件。

## 总结

实际的高速缓存替换策略试图通过预测高速缓存块的重新引用间隔来模拟最优替换。通常使用的LRU替换策略总是预测在未命中和命中时的近似重新引用间隔。然而，近实时重参考间隔的预测在丢失块的实际重参考间隔处于遥远的未来时低效地利用高速缓存。当应用程序引用的所有块的重新引用间隔在远处时，动态插入策略[25]通过保留高速缓存中的一些块来避免高速缓存抖动。然而，当由应用访问的块的重新引用间隔包括混合访问模式时，动态插入策略不能在高速缓存中保留具有近似重新引用间隔的块。本文表明，许多真实世界的游戏，服务器和多媒体应用程序展示这种混合访问模式。具体地，这样的应用经历对非时间数据（称为扫描）的引用的突发，其从高速缓存丢弃它们的活动工作集合。本文通过作出以下贡献来提高这些现实世界应用的性能：

1. 我们建议使用重参考间隔预测（RRIP）的高速缓存替换。 RRIP静态地预测所有丢失的高速缓存块的重参考间隔是在近即时重新引用间隔和远距离重新引用间隔之间的中间重参考间隔。 RRIP在重新参考时将重新参考预测更新为比先前预测短。 我们将此策略称为静态RRIP（SRRIP）。 我们显示SRRIP是抗扫描的，每个高速缓存块仅需要2位。
2. 我们提出两个SRRIP策略：SRRIP命中优先级（SRRIPHP）和SRRIP频率优先级（SRRIP-FP）。 SRRIP-HP预测接收到命中的任何高速缓存块将具有近即刻重新引用，因此应该在高速缓存中保持延长的时间段。SRRIP-FP另一方面预测经常引用的高速缓存块将有一个近乎即时再参考，因此它们应该在缓存中保留延长的时间段。 我们显示SRRIP-HP的性能明显优于SRRIP-FP，并且得出结论，扫描电阻不是从精确检测频繁引用的块，而是从防止接收命中的块被不接收命中的块（即扫描块）。
3. 我们建议动态RRIP（DRRIP）作为SRRIP-HP的增强。 DRRIP通过使用集合决斗来在插入具有中间重新引用间隔或具有远程重参考间隔的所有丢失的高速缓存块之间来提供扫描电阻和三线电阻。 除了SRRIP的硬件开销之外，除了单个饱和计数器之外，DRRIP不需要任何额外的硬件开销。

我们表明，在具有16路2MB LLC的单核处理器上，SRRIP和DRRIP的性能优于LRU平均4％和10％。我们还表明，SRRIP和DRRIP的性能优于LRU平均7％和9％ 具有16路8MB共享LLC的4核心CMP。 我们还显示，RRIP的性能优于迄今为止最先进的抗扫描替换算法LFU 2.5％。 对于正在研究的高速缓存配置，RRIP需要比LRU少2X的硬件和比LFU少2.5X的硬件。

在这项研究中，我们已应用重新参考间隔预测高速缓存未命中和学习缺失块的重新参考间隔没有任何外部信息。 在高速缓存命中的重新参考间隔预测理想地需要知道高速缓存块何时接收到其最后命中。 RIPP可以使用这样的信息来将重新引用的高速缓存块的重参考预测更新为中间，长或远的重参考间隔。 自动学习高速缓存命中的最后一个引用在没有任何外部信息的情况下更具挑战性。 在没有外部信息或存在死块和最后触摸预测器的情况下，高速缓存命中的预测重参考间隔[18,16]是我们正在进行的工作的一部分。