ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.12, No.4, February2016
E-mail: xsjl@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

面向预取的最优替换策略分析

李亚各,袁万腾

(中国航空计算技术研究所,陕西 西安 710065)

摘要:存储延迟通常是提高计算机系统性能的关键瓶颈。存储系统,尤其是最后一级高速缓存(LLC)是解决“存储墙”的有效方法,LLC的管理方法已经成为影响处理器性能的关键因素。预取技术能够利用时间局部性和空间局部性来减少流水线阻塞,从而提高计算机系统的整体性能。该文基于不同工作负载的特性来研究最先进的结合预取思想的LLC管理策略。我们实现了Bimodal Insertion Policy (BIP),该策略能够适应多变的工作负载。为了进一步减少cache的缺失率,我们使用Set Dueling策略在Static Re-Reference Interval Policy(SRRIP)和Bimodal Re-Reference Interval Policy(BRRIP)之间进行动态选择[13],原理是基于之前使用替换策略的历史信息。我们选择SPLASH-2作为测试基准来测试这些替换策略的性能。最后我们总结了不同策略的特性。

关键词: LLC; 替换策略; CRC

中图分类号:TP311 文献标识码:A 文章编号:1009-3044(2016)04-0089-04

1 介绍

最经典的替换策略是最近最少使用策略(LRU),但是在当前的多核处理器中,由于cache的高关联度会引起cache死块的问题,所以LRU策略逐渐失去了它的优势。同时,许多的工作负载中一个工作集就远远大于可用的cache大小。为了提高这些工作负载的性能,能够有效弥补LRU策略不足的动态替换策略引起了广泛的关注。

当目标cache写满的时候,替换策略选择该cache中某一行进行替换。在替换中,通常有两个关键的位置:最近最少使用位置(LRU)和最近最多使用位置(MRU) 。BIP替换策略的主要思想是:大部分情况下将新的cache行替换到LRU位置,剩余的情况替换到MRU位置。BIP策略能够适应工作集的变化并且同时保留了LRU策略的cache冲突保护机制。接下来我们使用Set Dueling机制在LRU策略和BIP策略之间动态选择一种更好地策略应用到之后的替换中。为了进一步提高系统的性能,我们实现了Dynamically Re-Reference Interval Policy (DRRIP),该策略使用Set Dueling机制在SRRIP策略以及BRRIP策略之间进行动态选择。DRRIP策略在每一个cache行设置一个计数器来存储这些行的Re-Reference Interval Prediction (RRPV)值[13]。

我们选择cache替换策略竞赛(CRC)提供的仿真环境来比较不同替换策略的优劣[12]。由于LLC的长延迟性以及LLC存取的低访问率,仿真环境对替换策略的算法的复杂性不做限制。我们通过选择典型的基准测试程序——SPLASH-2对替换策略进行评估,结果显示DIP策略能够提高12.64%的平均性能,DRRIP策略能够提高11.57% 的平均性能。

2 替换策略

2.1 DIP策略的分析

传统的LRU策略总是丢弃最近最少使用的数据,将新行插入到MRU行,这种做法会导致非LRU行的其他行的优先级得不到提高,即使有一些cache行刚被访问过。因为每当一个cache行被访问一次,所有的cache行优先级都会发生变化。BIP策略能够通过在低概率下将新行插入LRU位置来缓解这种情况,我们用参数ε来表示这个概率值[1]。我们设定ε= 1/32,即在32次插入新行的情况下,有31次插入MRU位置,1次插入LRU位置。

DIP策略通过在LRU策略和BIP策略之间进行动态选择来提高效率。考虑到工作负载的不同,DIP策略选择在标记组中发生缺失较少的策略来对新行进行替换操作。我们使用Set Dueling机制在没有增加显著硬件开销的情况下实现DIP[1]策略。假设在cache中有N组,每一组有若干行,K表示使用每种策略的组数($N=2^K$)。我们将cache分成K个区域,每个区域大小都为N/K,总共需要$\log_2$N bits,我们使用其中的$\log_2$K bits来标示选区,$\log_2$N/K bits来标示选区与第一组的偏移量[1]。DIP策略的存储开销仅需要2 bits来检测每组使用的替换策略。

对于本文中的提到的所有实验,我们设置一个cache中的组数N为16,K为4。我们使用2 bits来标示该组使用的替换策略是LRU或者BIP,组号为0以及5的倍数的组被标示为使用LRU策略,组号为3的倍数的组被标示为使用BIP策略。在cache中,辅助变量目录(ATD)与主变量目录(MTD)具有相同的关联度,我们将饱和计数器命名为策略选择器(PS),PS能够追

收稿日期:2015-01-15

踪ATD-LRU和ATD-BIP之间哪一个发生的缺失更少。如果ATD-LRU发生一次缺失，PS加1，相反地，如果ATD-BIP发生一次缺失，PS减1[1]。PS的最高有效位(MSB)能够指示哪一种策略发生更少的缺失。DIP策略的原理图如图1所示。

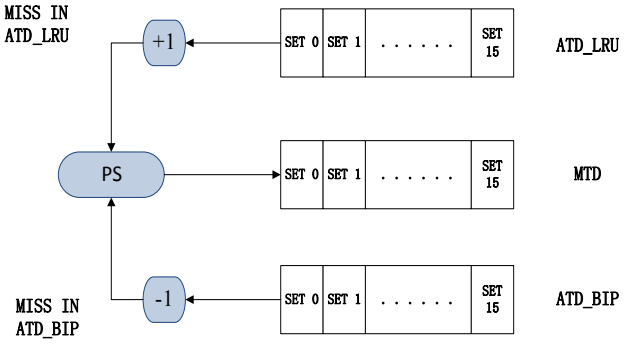

图1　DIP策略的原理图[1]

在MTD中共有16组，从0号组到15号组，我们选择其中四组来标示使用LRU策略：0号、5号、10号、15号；再选择四组来标示使用BIP策略：3号、6号、9号、12号；剩余的组所使用的替换策略是由PS选择出来的效果较好的一种策略。如上文所述，如果ATD-LRU发生一次缺失，PS加1，相反地，如果ATD-BIP发生一次缺失，如果PS的MSB等于1，那么MTD使用BIP策略，否则使用LRU策略。

2.2 DRRIP策略的分析

RRIP值代表预测某一cache块被重新调用的次序[13]。在RRIP链的前端，表示该cache块很快会被再次调用，而在RRIP链的后端则表示该cache块被再次调用的间隔会很久。DRRIP策略使用Set Dueling机制在SRRIP策略和BRRIP策略之间动态选择一种更好地策略应用到下一次替换中，以此来减少发生缺失的次数。

1)SRRIP策略

如果一个cache块间隔很久才会被调用，那么就有可能引起cache的高缺失率，低cache的使用率。为了阻止那些再次调用间隔很久的cache块污染cache，SRRIP策略在每个cache块中使用M bits来存储该块的RRPV值。RRPV值会随着每个块被调用的频率动态变化。对于本文中的提到的所有实验，我们设置M的值为2。RRPV值等于‘0’，说明该cache块最近被调用过，而且预测处理器很快会再次调用该cache块；RRPV值等于‘3’，说明该cache块最近没有被调用过，而且处理器在短时间内不会再次调用该cache块；RRPV值等于‘1’或者‘2’时，说明该cache块会有较高的可能性被调用。

在初始情况下，所有的cache块的RRPV值都等于‘3’。一旦cache命中，该cache块的RRPV值就被设置成‘0’；相反地，一旦cache发生缺失，SRRIP策略会选择再次调用间隔很久的cache块进行替换，首先查找RRPV值为‘3’的cache块进行替换，并将其RRPV值设置成‘2’，如果没有找到RRPV值为‘3’的cache块，就将所有cache块的RRPV值加1后再进行查找，直到找到RRPV值为‘3’的cache块。图2表示的是四路初始值为‘1’的cache块，在一系列混合存取模式下的SRRIP策略的工作流程。

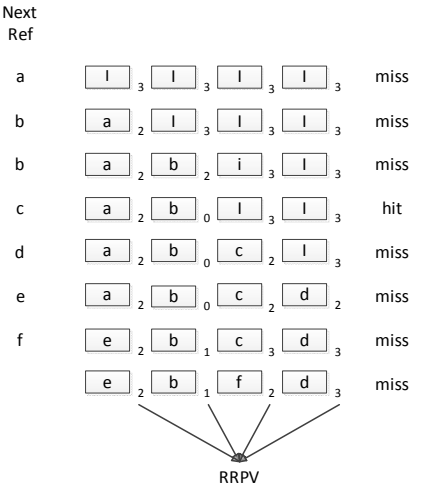

图2　混合存取模式下的SRRIP策略工作流程

从图2中我们可以清楚地看到SRRIP策略的工作过程。SRRIP策略能够自动适应工作集中应用程序的大小，并能够明显降低cache缺失率，提高系统效率。

2)DRRIP策略

为了避免cache冲突以及适应不同种类的工作集，参考文献[13]提出了BRRIP策略。与上文提到的BIP策略类似，BRRIP策略在低概率下将新cache块替换到RRPV值为‘2’的块，我们用参数ε来表示这个概率值，设定ε = 1/32，即在32次插入新行的情况下，有31次插入RRPV值为‘3’的位置，1次插入RRPV值为‘2’的位置。

但是对于不存在cache冲突的存取模式，总是使用BRRIP策略是有害无益的。所以我们通过使用Set Dueling机制在SRRIP策略和BRRIP策略之间进行动态选择，基于历史信息选择发生较少缺失的策略应用到其他cache块。具体实现方法类似于上文描述的DIP策略。DRRIP策略的存储开销仅需要2 bits来检测每组使用的替换策略。DRRIP策略的原理图如图3所示。

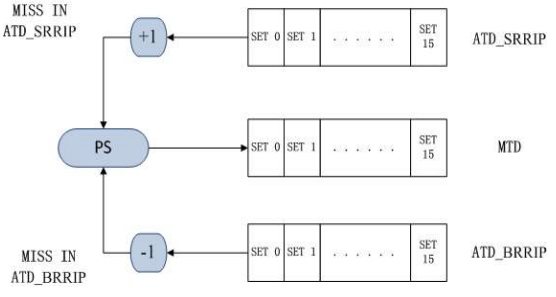

图3　DRRIP策略的原理图[1]

对于本文中的提到的所有实验，我们设置一个cache中的组数N为16，K为4。PS能够追踪ATD-SRRIP和ATD-BRRIP之间哪一个发生的缺失更少。如果ATD-SRRIP发生一次缺失，PS加1，相反地，如果ATD-BRRIP发生一次缺失，PS减1。PS的MSB能够指示哪一种策略发生更少的缺失。如果PS的MSB等于1，那么MTD使用BRRIP策略，否则使用SRRIP策略。

3 实验方法

3.1 仿真环境

为了比较不同替换策略的优劣，我们选择cache替换策略竞赛(CRC)提供的仿真器[12]。CRC是基于跟踪驱动的仿真器，

提供了类结构中的元数据和函数，用来在LLC中选择被替换的cache行。在仿真过程中，可以选择单核模式或者多核模式。每一个cache行有8 bits，全局内存为1 Kb。

3.2 CRC配置参数

CRC中的RADEME.txt文件对存储结构的具体配置进行了详细的描述。LLC的配置包括16路的组，64 bytes的行，以及每个核有1 MB的大小。在单核中，LLC大小为1MB，在四核中，有4MB的共享LLC。L1 cache的大小为32KB, L2 cache的大小为256KB。LLC的具体配置参数如表1所示。

表1　Cache配置

Cache Size	Line Size	Associativity	Sets	Threads
1024KB	64B	16	1024	1

在实验中，单核以及双核的配置均为三级cache，我们主要研究LLC的cache缺失率。我们通过选择典型的基准测试程序——SPLASH-2来对替换策略进行评估。

4 测试结果与实验分析

4.1 实验结果分析

Cache缺失率能够表明存储系统的性能以及平均内存访问时间，所以cache缺失率是评估一种替换策略优劣的重要指标。本文基于处理器的单核模式来比较每一种替换策略的优缺点。在我们使用的仿真器中，提供两种基本的替换策略：LRU策略和RANDOM策略。我们基于仿真器CRC实现了DIP策略和DRRIP策略。实验结果如表2所示。

表2　不同替换策略的cache缺失率比较

Benchmark\ Cache miss rate	LRU	RANDOM	DIP	DRRIP
Ls	99.673	99.673	99.673	99.673
FFT	98.6264	98.6264	98.6264	98.6264
RADIX	83.2914	71.888	69.256	70.853
BARNES	72.8731	68.2726	66.736	66.9067

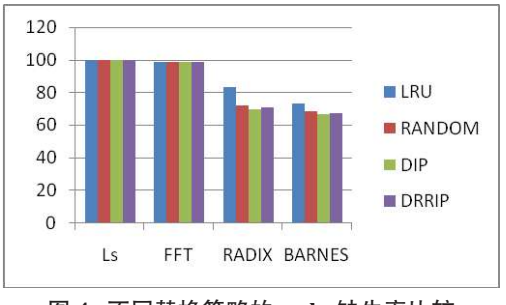

图4　不同替换策略的cache缺失率比较

图4表明我们实现的DIP策略以及DRRIP策略的cache缺失率比传统的LRU策略和RANDOM策略有明显的降低。通过实验数据我们可以得到，在最好的情况下，DIP策略降低了16.85%的cache缺失率，DRRIP策略降低了14.94%的cache缺失率。能够减小cache缺失率的根本原因在于，DIP策略和DRRIP策略能够克服cache空间的无效占用，将预测到很久不再调用的cache行替换掉，提高了cache空间的使用率。LRU策略的高cache缺失率主要由两个原因导致：首先，如果某一行不存在时间局部性就意味着这一行可能很久都不会再被调用，如果将这样的行保留在cache内，显然是毫无意义的；另外，如果某一行再次被调用的距离大于cache块的大小，那么LRU策略就会再该行被调用之前将它替换掉。为了克服LRU策略的缺点，我们实现了DIP策略和DRRIP策略通过使用Set Dueling机制来动态的选择一种最优策略，其中用到了预测机制，对某一cache行将来会被调用的可能性进行预测。

以上的实验结果显示DIP策略能够提高12.64%的平均性能，DRRIP策略能够提高11.57%的平均性能。这些数据表明我们实现的LLC替换策略能够有效地避免cache污染，适应多种不同的工作负载，显著地提高了cache性能。同时，结果表明，对于不同的测试基准程序每种替换策略得到的cache缺失率也不相同，这主要是因为能够优化的空间与特定测试基准程序有密切关系。

4.2 硬件开销

BIP策略在低概率ε下将新行插入MRU位置，我们设置ε = 1/32，因此需要5 bits计数器来控制新的插入行要插入MRU位置或是LRU位置。DIP策略中的PS计数器需要10 bits来对LRU策略和BIP策略进行选择。实现Set Dueling机制仅需一个单饱和计数器。SRRIP策略的实现需要对每个cache块增加一个2 bits的寄存器来存储RRPV值（‘0’，‘1’，‘2’，和‘3’），4个逻辑单元来进行查找RRPV值为‘3’的cache块，如果没有找到，就需要额外的一个逻辑对组中所有RRPV寄存器进行加1操作。BRRIP策略所需的存储开销在SRRIP策略的基础上，再加上一个额外的5 bits计数器用来控制将新cache块替换到RRPV值为‘2’的块。DRRIP策略是对SRRIP策略和BRRIP策略进行动态选择，因此需要10 bits的PS计数器。

从上述的比较中，我们可以考出DRRIP策略比DIP策略需要较多的开销。然而这些开销没有对关键路径进行改变，因此不会对cache的存取时间造成很大的影响。

5 总结

在本文中，我们实现了两种基于预取思想的LLC替换策略：DIP策略和DRRIP策略。这两种策略动态的选择一种发生缺失较少的策略来对大部分cache行进行替换操作，克服了典型的LRU策略的缺点：总是替换最近最少使用的cache行，即使该行可能很快被再次调用。我们选择仿真器CRC来比较不同替换策略的优劣，测试基准为标准测试集为SPLASH-2中的FFT，RADIX和BARNES。从实验结果可以看出，DIP策略和DRRIP策略的cache缺失率明显低于传统的LRU策略和RANDOM策略。大量的实验数据表明在单核模式下，cache缺失率可以从88.29%降到69.26%。同时，我们对两种策略的硬件开销进行了对比，DRRIP策略的开销略大于DIP策略的开销，由于对关键路径没有影响因此可以忽略不计。从实验结果得出，不同的替换策略针对不同的测试基准会有不一样的优化，我们可以根据程序特性，通过选择平均性能最优的替换策略来获得最高的性能提升。

参考文献：

[1] Subramanian R.Adaptive Caches: Effective Shaping of Cache Behavior to Workloads[C]. 1995.

[2] Zhao Jianmin, Zhao Jianhua. An Optimal Replacement Policy for a System with Finite Spares[J].Journal of Systems Engineering and Electronics, 1999.

[3] Steven P Vanderwiel, David J Lilja, Data Prefetch Mechanisms[C].ACM Computing Surveys (CSUR), 2000.

[4] Wong W A, Baer J L. Modified LRU Policies for Improving Second-level Cache Behavior[C].ACHPCA-6, 2000.

[5] Zhuang Weiqiang, Hu min, Wang Dingxing.Replacement Policy for Caching World-Wide Web Documents Based on Site-Graph Model[C].Tsinghua Science and Electronics, 2001.

[6] Lin W, Fide C, Falsafi B. Predicting Last-touch References under Optimal Replacement[R].Technical Report of Michigan University, 2002.

[7] Megiddo N,Modha D S.ARC: A Self-tuning, Low Overhead Replacement Cache[C].Proceeding of the 2nd USENIX Conference on File and Storage Technologies, 2003.

[8] Qureshi M K, Thompson D, Patt Y N. The V-Way Cache: Demand Base Associativity via Global Replacement[J].ISCA, 2005(32):544-555.

[9] Nai Zhengbian, Hao Chen.A Least Grade Page Replacement Algorithm for Web Cache Optimization[C].Proceedings of the First International Workshop on Knowledge Discovery and Data Mining, 2008.

[10] Wang Deli, Gao Deyuan.A Sharing-Aware Active Pushing Cache Technology on Chip-Multiprocessor [J].Journal of Xi'an Jiao Tong University, 2010, 44(10):18-23.

[11] Kahkashan Tabassum, Asia Sultana, Damodaram.A.A Heuristic Cache Replacement Policy for Data Caching[C].Proceeding of 2010 4th International Conference on Intelligent Information Technology Application, 2010.

[12] Yuval Peress, Ian Finlayson, Dr Gary Tyson, CRC: Protected LRU Algorithm[C].JILP Worshop on Computer Architecture Competitions, 2010.

[13] Jaleel A, Theobald K B, Jr Steely S C.High Performance Cache Replacement Using Re-reference Interval Prediction (RRIP) [C].Proceedings of the 38th International Symposium on Computer Architecture, 2010.

[14] Carole Jean Wu, Aamer Jaleel Margaret Martonosi, Simon C Steely. "PACMan: Prefetch-Aware Cache Management for High Performance Caching[C].Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture, 2011.

[15] Li Chongming, Wang Dongshen, Xue Yibo. Enhanced Adaptive Insertion Policy for Shared Caches[C].Proceedings of the 9th international Conference on Advanced Parallel Processing Technologies, 2011.

[16] Tan Lin, Yang Jian, Cheng Zhijun. Optimal Replacement Policy for Cold Standby System[J[. Chinese Journal of Mechanical Engineering, 2011.

[17] Shan Ding,Li Yuanyuan. LRU2_MRU Collaborative Cache Replacement Algorithm on Multi-core System [C].Proceedings of 2012 IEEE International Conference on Computer Science and Automation Engineering, 2012,23(1):14-17.