doi:10.3969/j.issn.1673-9833.2020.04.008

改进的偏向写调度的混合内存缓冲区调度策略

刘 兵[1,2]，汪令辉[3]，张 锐[2]，崔 莹[2]，段 峰[2]

（1. 中国科技大学 计算机科学与技术学院，安徽 合肥 230027；2. 铜陵职业技术学院 信息工程系，安徽 铜陵 244061；
3. 铜陵有色金属集团公司，安徽 铜陵 244000）

摘 要: 提出一种新的相变存储器（PCM）和DRAM混合内存偏向写调度缓冲区页面调度策略（FWLRU）。该策略根据PCM和DRAM在读上区别不大，所以只进行“写”热页调度进入DRAM，而不专门进行“读”热页的调度，同时在PCM和DRAM的页面置换时，不淘汰页面，采取互换的原则，以增加页面的命中率和减少PCM的写。实验结果表明，该策略能有效提高页面的命中率和减少PCM的写。

关键词：非易失存储器；相变存储；混合内存；缓冲区调度

中图分类号：TP333 **文献标志码**：A **文章编号**：1673-9833(2020)04-0048-06

引文格式：刘 兵，汪令辉，张 锐，等 . 改进的偏向写调度的混合内存缓冲区调度策略 [J]. 湖南工业大学学报，2020，34(4)：48-53.

An Improved Hybrid Memory Buffer Scheduling Strategy with Write Preference

LIU Bing[1,2]，WANG Linghui[3]，ZHANG Rui[2]，CUI Ying[2]，DUAN Feng[2]

（1. School of Computer Science and Technology，University of Science and Technology of China，Heifei 230027，China；
2. Department of Information Engineering，Tongling Vocational and Technical College，Tongling Anhui 244061，China；
3. Tongling Nonferrous Metals Group Co.，Tongling Anhui，244000，China）

Abstract：A proposal has been made of a new hybrid memory scheduling strategy (FWLRU) for phase change memory (PCM) and DRAM. Based on the difference between PCM and DRAM in reading, this proposed strategy only performs "write" hot page scheduling to enter DRAM instead of "read" hot page scheduling. Meanwhile, pages will not be eliminated in the page replacement of PCM and DRAM, with the principle of exchange adopted to increase the hit rate of pages and reduce PCM writing. The experimental results show that the strategy can effectively improve the hit rate of pages and reduce PCM writing.

Keywords：non-volatile memory；phase change memory；hybrid main memory；buffer scheduling

随着大数据、人工智能等技术的发展，数据的处理量越来越大，这对计算机的主存储器提出了越来越高的要求。当前，以动态随机存取存储器（dynamic random access memory，DRAM）为主的主存储技术也面临着较大的挑战，主要面临的问题为DRAM存储集成已达极限，且能耗也是一个重要问题。人们从软件和硬件等方面提出了多种方式，希望弥补这一缺点，比如通过非易失存储（non-volatile memory，NVM）来解决这一问题。非易失存储具有较高的集成度、非易失、低能耗、字节寻址等特点。非易失存储器有电阻随机存储器ReRAM[1]、自旋磁存储器[2]、相变存储器[3]（phase change memory，PCM）等。其中，

收稿日期：2019-11-25

基金项目：安徽省高校自然科学研究基金资助重点项目（KJ2018A0749，KJ2018A0751）

作者简介：刘 兵（1974-），男，安徽肥东县人，中国科技大学访问学者，铜陵职业技术学院高级工程师，副教授，主要从事数据库技术和大数据方面的研究，E-mail：yg_liu@163.com

PCM 具有良好的可扩展性，有望成为新一代的主流技术。

1　研究背景和相关工作

1.1　PCM 的相关特性

相变存储器是一种硫族化合物，分为晶体状态和非晶体状态。它具有写的不对称性，PCM 写 1（SET）的时候，要施加一个时间长、电压低的电脉冲，温度在结晶点以上、融化点以下，导致其结晶，物质从非晶态到晶态转化。PCM 写 0（RESET）的时候，要施加一个电压高、时间短的电脉冲，当温度上升到溶点后，再经过一个淬火（降温速率大于 109 K/s）的过程，物质从晶态到非晶态转化。

相变存储器在晶态和非晶态时，其阻抗是不同的，当施加一个电压时，对应的电流不同，从而可以判断为 0 或 1。PCM 读的过程中，电流通过时产生的热量很小，不会引起晶态的变化。

PCM 和 DRAM 相比较，具有存储密度大和无空闲功耗的优点。DRAM 的工艺制程是 PCM 的 4 倍，所以在相同的芯片面积下，PCM 可以增加更大的容量；DRAM 空闲时，需要通过不断地刷电来保持数据，而 PCM 为非易失性存储设备，不需要刷电。

PCM 和 DRAM 相比较，也有它自身的缺点，主要如下：

1）PCM 的读操作时间是 DRAM 的 2 倍，功耗相差不大；

2）PCM 的写操作时间是 DRAM 的 10 倍，PCM 写功耗比 DRAM 大；

3)PCM 的写次数为 108~1 012，DRAM 几乎无限，写次数为大于 1 015。

1.2　PCM 在内存中的应用

1.2.1　混合内存架构

目前，针对 PCM 的特性，许多研究者尝试结合二者的优点，提出了混合内存 [4–9] 的概念。混合内容的主要架构方式包括层次架构和平行架构两种。

1）层次架构。该方式下，DRAM 作为 PCM 的缓存，所有的请求都先访问 DRAM，当请求没有选中是时，再访问 PCM。这种架构方式的优点是利用 DRAM 弥补 PCM 的写不足，同时利用 PCM 增加存储密度，利用其非易失性存储特点，减少空闲能耗。

2）平行架构。该方式中 PCM 和 DRAM 的同一层混合使用，整个内存由两者共同组成，统一编址，数据只放在 PCM 或者 DRAM 中的一个中，如将写倾向较高的页面放在 DRAM 中，将读倾向较高的页面放在 PCM 中。

在层次架构下，当出现对内存需要较大的调用时，会增大 DRAM 和 PCM 交换的工作量，系统效率较大下降，并且会增大 PCM 的磨损。平行架构中，根据 PCM 也是字节寻址的特点，将 DRAM 和 PCM 统一地址空间，但由于二者的读写等特性不一样，为了降低能耗并延长 PCM 的使用时间，缓冲区的管理算法就显得尤为重要。

1.2.2　平行架构缓冲区管理算法

已有关于混合内存的缓冲区管理算法主要有 LRU-WPAM（least recently used with prediction and migration）[10–11] 和 CLOCK-DWF（clock with dirty bit and write frequency）[12]。其中，CLOCK-DWF 算法对 CLOCK 算法进行了改进，通过记录每个页面的写次数来判断读倾向和写倾向，从而调度写倾向在 DRAM 中，读倾向在 PCM 中；LRU-WPAM 算法以最近最少使用（LRU）算法为基础，增加了一个页面读写预测机制，从而进行页面的调度。

在 LRU-WPAM 的缓冲区管理中，当缓冲区页面未命中时，与 LRU 算法一样，选择缓冲区最近最少使用的页面进行置换；当缓冲区中页面命中时，首先根据读写请求修改页面权值，然后判断是否达到阀值，并根据权值决定是否将页面移动到 PCM 或者 DRAM 中，如果目标存储器上没有空闲空间，在 DRAM 中选择读子队列尾部页面释放，在 PCM 上选择写子队列尾部页面释放。

在 CLOCK-DWF 的缓冲区管理中，首先，将 DRAM 和 PCM 分别组成一个环状队列；然后，根据数据访问时的读写类型，将写请求的页面存放在 DRAM 中，读请求页面存放在 PCM 中。当 DRAM 空闲空间不足时，将“写”冷页写入 PCM，当 PCM 空闲空间不足时，用 CLOCK 页面调度的算法对页面进行调度。

2　FWLRU 混合内存缓冲区调度策略

在上述算法的混合内存管理中，根据“写”热页和“读”热页的判断，调度页面在 PCM 或者 DRAM 中存储。如果所在页面当前存储和判断的结果不一致，则需要进行 PCM 和 DRAM 的页面迁移。根据以上调度算法，主要会造成以下几个问题：

1）频繁地在 PCM 和 DRAM 中进行页面迁移，要消耗大量的系统资源。同时，当 PCM 和 DRAM 中的空闲空间不足时，需要从二者中选择页面进行释放，释放的页面可能就是即将访问的页面，这样就会

将缓冲区原先命中的访问变成没有命中，造成缓冲区命中率下降。

2）将“读”热页从 DRAM 中迁入 PCM，迁移写入时实际上增加了 PCM 的写，与减少 PCM 写的初衷不一致。

3）当页面的访问是读倾向较多的时候，按照 LRU-WPAM、CLOCK-DWF 算法的要求，都要迁移进入 PCM，这样反而造成 PCM 写的增加。

另外，根据 A. R. Alameldeen 等的研究 [13-15]，内存数据访问具有明显的局部性，局部数据的访问达到 40% 以上，有些情况下甚至超过 60%。

针对以上情况，本文提出一种偏向写调度缓冲区调度策略（favors write least recently used，FWLRU）的混合内存缓冲区调度策略，主要进行了以下两个方面的优化：

1）只进行“写”热页的调度，而不进行“读”热页的调度。混合内存的主要目的是充分利用 PCM 的高存储密度和低能耗这两个优点来优化内存，避免 PCM 读写不均衡、写有限等缺点。DRAM 和 PCM 在读上区别并不大，如果强制将所有大于权值的读倾向页移动进入 PCM，将增加 PCM 的写操作，特别是如果读操作较多时，写的次数将更多。但是如果不进行读页面的调度，对系统性能将没有什么影响。所以在 FWLRU 算法中只考虑将“写”热页调度进入 DRAM，而不进行“读”热页的调度。

2）“写”热页采取 PCM 和 DRAM 页面互换的原则进行迁移。在将“写”热页从 PCM 置换到 DRAM 过程中，当 DRAM 空间不够时，如果采取淘汰策略，有可能淘汰的页面就是下一次就要访问的页面，这样就把将要访问的页面淘汰掉了，从而把原本命中的操作变成了不命中，降低了命中率。而 FWLRU 算法中，不进行页面的淘汰，只是将 PCM 页面的 DRAM 中的页面采取互换原则，以此避免淘汰可能将要被访问的页面，提高页面的命中率。

2.1 访问行为记录

FWLRU 混合内存缓冲区调度策略中使用了 3 个 LRU 队列：DRAM 读队列（DR）、DRAM 写队列（DW）、PCM 写队列（PW）。

LRU 队列在缓冲区中按照访问时间组成队列，其中，最近访问位于队列首部，最长时间访问位于队列尾部。DR 队列记录 DRAM 中页面的读操作，DW 队列记录 DRAM 中的写操作，PW 队列记录 PCM 中的写操作，都按照最近访问原则从前至后排列。

2.2 PCM 中“写”热页和 DRAM 中页面的互换

FWLRU 偏向写调度的缓冲区调度策略中，当缓冲区块命中时，修改页面的权值，如果是“读”命中，权值增加；如果是“写”命中，权值减少。将权值和“读写热页判定标准”阀值进行比较，若页面权值小于阀值，则说明是一个“写”热页。根据缓冲区和内存空间的物理地址（internal memory address）映射，查看这个页面的物理地址是在 DRAM 中还是在 PCM 中，如果在 PCM 中，要进行 PCM 和 DRAM 空间页面的互换，同时将页面加在 LRU 队列首部。和“读写热页判定标准”阀值比较，如果大于阀值，则说明是一个“读”热页，不进行操作。

同时，根据读写操作，如果是写，那么根据物理地址的映射，加入到 DW 或者 PW 队列的头部；如果是 DRAM 读写，那么加入到队列的头部，并修改页面的权值。

“写”热页将执行页面从 PCM 写入 DRAM，如果 DRAM 空间有空闲，那么直接写入；如果 DRAM 空间已满，那么将 DR 队列尾部页面和“写”热页互换，如图 1 所示。页面互换时，不进行 DRAM 页面的淘汰，这样可以避免将有可能即将访问的页面淘汰掉，把页面原本的命中操作变成不命中操作，从而降低缓冲区的页面命中率。

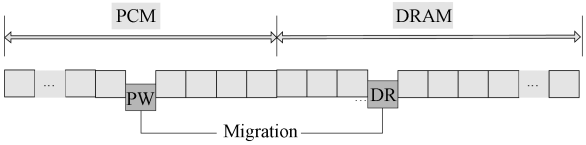


图 1 PCM 写热页与 DRAM 中 DR 队列尾部页面互换

Fig. 1 Exchange of PCM hot page and DR queue tail page in DRAM

2.3 调度流程

算法 1 描述了混合内存缓冲区调度策略的调度流程。算法中 IMAddr（internal memory address）为页面从缓冲区映射到内存的物理地址，物理地址根据字节地址范围划分为 DRAM 和 PCM 区域。页面如果判定是“写”热页，且 IMAddr 在 PCM 中时，执行 2.2 所述页面互换工作。缓冲区页面访问类型为“读”时，修改页面权值加 1，同时将页面序列加入到 LRU 和 DR 队列头部。

算法 1 混合内存缓冲区调度策略流程

```
If p is a write Request then
        P.Authority++;
        Move p to LRU position of F;
        If p. Authority>threshold then
    If IMAddr is in PCM then
            Move p to PW position F;
            PW.AccessCount++;
```

```
                If DRAM is not full then
                        Move p to DRAM;
                else
                        Migration IMAddr and Dr queue tail page in PCM&DRAM;
                        Move p to DW position of F;
                    return IMAddr;
                Endif;
            Endif;
            If IMAddr is in DRAM then
                DW.AccessCount++;
                Move p to DW position of F
                retuen IMAddr;
            Endif;
        Endif;
        If p is a read Request then
            P.Authority--;
            Move p to LRU position of F;
            If IMAddr is in DRAM then
                Move p to DR position of F;
            Endif;
        Endif;
        End.
```

算法完成后，返回 IMAddr 值。

3　实验结果及分析

3.1　实验方法

本实验采用在 ubuntu 18.04 系统上架设 GEM5[16] 模拟器仿真，同时安装 NVMain[17-18] 模拟相变存储器，从而实现 DRAM 和 PCM 混合实验环境。系统采用 SE（系统调用）模式，PCM 的延迟数据参照 F. Bedeschi 的研究 [19]，DRAM 的延迟参照 Micron 的测试 [20]。每个页面大小设定为固定值 4 kB，DRAM 和 PCM 的比例采取固定配置形式，按照 1∶4 配置，整体存储空间按照实验需要进行增大或减少。

本实验选用的数据集由真实数据和合成数据两部分构成，其中真实数据出自于安徽芜湖某电商网站交易系统的某日交易记录，该数据集对数据库进行了 356 733 次读和 115 790 次写操作；合成数据来自开源软件 DiskSim，版本为 4.0。DiskSim 进行磁盘读写模拟，同时改变配置参数，设置局部性读写不同的比例来得到系列数据集，挑选其中具有代表性的 4 个数据组参与混合内存实验。局部性（Locality）中“80%/20%”，指 80% 的数据发生在 20% 的空间上。具体实验数据集测试参数见表 1。

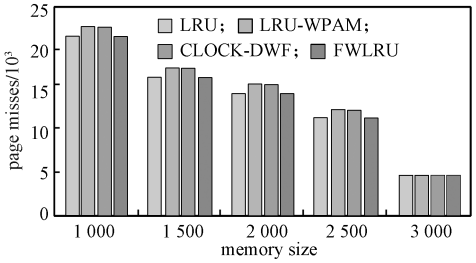
表 1　实验数据集

Table 1　Experimental data set

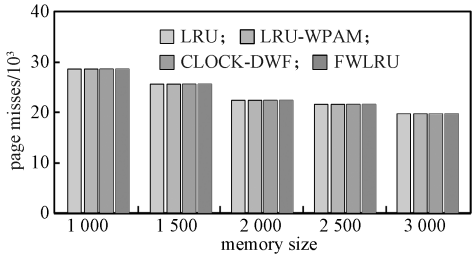
Type	Total Reference	Read/Write Ratio	Locality	Total Request
Trace9151	10 000	90%/10%	80%/20%	50 000
Trace8987	10 000	90%/10%	50%/50%	50 000
Trace3377	10 000	50%/50%	80%/20%	50 000
Trace1899	10 000	10%/90%	80%/20%	50 000
OLTP	10 000	75%/25%		472 523

3.2　页面命中率

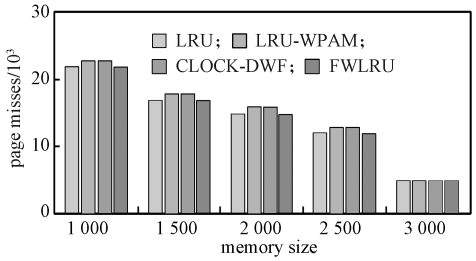
将 FWLRU 混合内存缓冲区调度策略和 LRU、LRU-WPAM、CLOCK-DWF 在模拟器上进行了页面命中率检测。图 2 给出了 5 组测试数据集在内存页面逐渐增大的情况下，4 种不同的缓冲区调度下页面未命中的数量。



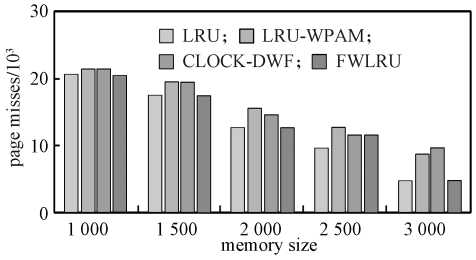
a）Trace9151



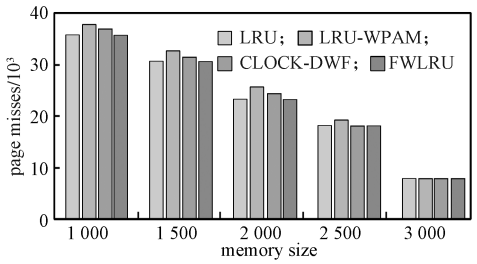
b）Trace8987



c）Trace3377



d）Trace1899



e）OLTP

图 2 不同调度策略下的命中率实验结果

Fig. 2 Experiment results of hit rate under different scheduling

分析图 2 所示实验结果数据，可得出以下结果：

1）从图 2a 至 2e 的 5 组类型数据的实验情况看，随着内存容量逐渐增大，页面未命中的数量明显减小。可见内存容量的大小对命中率有着直接的影响，容量越大，命中率越高。

2）从命中率情况看，FWLRU 算法的命中率比 LRU-WPAM 和 CLOCK-DWF 的要高，接近 LRU 算法的。这和 FWLRU 算法中不进行页面的淘汰有关，只是将“写”热页从 PCM 到 DRAM 的互换，防止了将即将访问的页面淘汰而造成的命中率下降。

通过实验结果对比可以看出，相对于 LRU-WPAM 和 CLOCK-DWF 算法，FWLRU 算法提高了页面的命中率。

3.3 PCM 写次数

设定内存空间大小固定为 1GB（DRAM 与 PCM 的比值为 1∶4）的情况下，对 PCM 在 FWLRU、LRU、LRU-WPAM、CLOCK-DWF 4 种策略下的写总次数进行实验模拟，结果如图 3 所示。

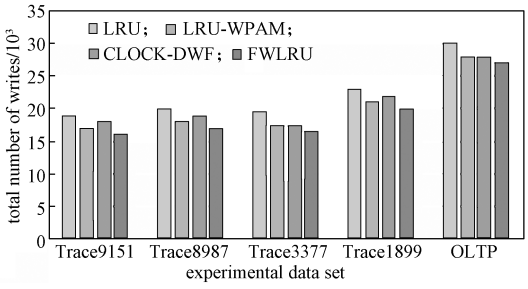


图 3 不同调度策略下的 PCM 写次数实验结果

Fig. 3 Experiment results of PCM write times under different scheduling

通过图 3 所示实验数据可以看出，LRU 由于不考虑两种存储介质的不同，没有进行“写”热页和“读”热页的调度，所以 PCM 写的次数最多；LRU-WPAM 和 CLOCK-DWF 考虑了 PCM 和 DRAM 这两种存储介质，并进行了将“写”热页存放在 DRAM、“读”热页存放在 PCM 上，导致 PCM 写的数量有所降低，但将“读”热页集中到 PCM 上，移动的过程也增加了 PCM 的写，并且对读比例大的页面进行访问，反而有可能加大 PCM 的写。FWLRU 考虑读操作上 DRAM 和 PCM 的区别不大，故不进行“读”热页的移动，只将“写”热页互换到 DRAM 中，所以 PCM 写的次数比 LRU-WPAM 和 CLOCK-DWF 的有所降低。实验结果证明，FWLRU 对 PCM 进行了优化，写的次数减少。

4 结论

非易失性存储材料 PCM 是解决 DRAM 的存储密度和降低能耗的好材料，但是 PCM 具有读写不对称和写有限等缺点，许多研究者提出以多种混合内存的缓冲区调度策略来解决这一问题。本文在混合内存架构的基础上，提出了一种偏向写调度的混合内存缓冲区调度策略。

1）只将“写”热页从 PCM 置换进 DRAM，没有将“读”热页写入 PCM，避免了调度过程中或者频繁调度中对 PCM 的写，特别是读偏向较多的页面；

2）PCM 和 DRAM 采取页面互换的形式，不从 DRAM 中淘汰页面，避免了将可能即将访问的页面淘汰，将命中操作变成没有命中操作的情况，提高了页面访问的命中率。

3）本文从缓冲区调度入手来解决 PCM 读写不对称、写有限等问题。但是冷热页的划分还是采取简单的权值计数方式，这种冷热页划分方式是否过于简单还有待考证；而且没有对 PCM 的磨写均衡进行考虑，这是下一步研究的方向。

参考文献：

[1] MAO Manqing，CAO Yu，YU Shimeng，et al. Optimizing Latency，Energy，and Reliability of 1T1R ReRAM Through Appropriate Voltage Settings[C]// Processing of the 33rd IEEE International Conference on Computer Design(ICCD 2015). Piscataway：IEEE，2015，359-366.

[2] KULTURSAY E，KANDEMIR M，SIVASUBRAMANIAM A，et al. Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative[C]//2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). Piscataway：IEEE，2013：256-267.

[3] 张鸿斌，范 捷，舒继武，等 . 基于相变存储器的存储系统与技术综述 [J]. 计算机研究与发展，2014，51(8)：1647-1662.

ZHANG Hongbin，FAN Jie，SHU Jiwu，et al. Summary of Storage System and Technology Based on Phase Change Memory[J]. Journal of Computer Research and Development，2014，51(8)：1647−1662.

[4] YANG Jun，WEI Qingsong，CHEN Cheng，et al. NV-Tree：Reducing Consistency Cost for NVM-Based Single Level Systems[C]//Processing of the 13th USENIX Conf on File & Storage Technologies(FAST 2015). Berkeley：USENIX Association，2015：167−181.

[5] OUKID I，LASPERAS J，NICA A，et al. FPTree：A Hybrid Scm-Dram Persistent and Concurrent B-Tree for Storage Class Memory[C]//Processing of International Conference (SIGMOD 2016). New York：ACM，2016：371−386.

[6] LEE S，LIM K，SONG H，et al. WORT：Write Optimal Radix Tree for Persistent Memory Storage Systems[C]// Processing of the 15th USENIX Conference on File and Storage Technologies(FAST 2017). Berkeley：USENIX Association，2017：257−270.

[7] WANG C，WEI Q，WU L，et al. Persisting RB-Tree into NVM in a Consistency Perspective[J]. ACM Transactions on Storage，2018，14(1)：1−27.

[8] CHI Ping，LEE Wangchien，XIE Yuan. Adapting B+-Tree for Emerging Non-Volatile Memory-Based Main Memory[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2016，35(9)：1461−1474.

[9] HWANG D，KIM W，WON Y，et al. Endurable Transient Inconsistency in Byte-Addressable Persistent B+-Tree[C]// Processing of the 16th USENIX Conference on File and Storage Technologies(FAST 2018). Berkeley：USENIX Association，2018：187−200.

[10] SEOK H，PARK Y，PARK K W，et al. Efficient Page Caching Algorithm with Prediction and Migration for a Hybrid Main Memory[J]. ACM SIGAPP Applied Computing Review，2011，11(4)：38−48.

[11] SEOK H，PARK Y，PARK K H. Migration Based Page Caching Algorithm for a Hybrid Main Memory of DRAM and PPAM[C]//Proceedings of the 2011 ACM Symposium on Applied Computing. TaiChung：ACM，2011：595−599.

[12] LEE S，BAHN H，NOH S H. CLOCK-DWF：A Write-History-Aware Page Replacement Algorithm for Hybrid PCM and DRAM Memory Architectures[J]. IEEE Transactions on Computers，2014，63(9)：2187−2200.

[13] ALAMELDEEN A，WOOD D. Frequent Pattern Compression：A Significance-Based Compression Scheme for L2 Caches[R]. USA：University of Wisconsin-Madison Department of Computer Sciences，2004：1−14.

[14] ARJOMAND M，JADIDDI A，SHAFIEE A，et al. A Morphable Phase Change Memory Architecture Considering Frequent Zero Values[C]//Proceedings of IEEE International Conference on Computer Design(ICCD’11). Amherst：IEEE Press，2011：373−380.

[15] YANG J，ZHANG Y，GUPTA R. Frequent Value Compression in Data Caches[C]//Proceedings of 33rd Annual ACM/IEEE International Symposium on Microarchitecture(MICRO’00). Monterey：ACM Association Press，2000：258−265.

[16] BINKERT N，BECKMANN B，BLACK G，et al. The Gem5 Simulator[J]. ACM SIGARCH Computer Architecture News，2011，39(2)：1−7.

[17] POREMBA M，XIE Y. Nvmain：An Architectural-Level Main Memory Simulator for Emerging Non-Volatile Memories[C]//Proceedings of the 2012 IEEE Computer Society Annual Symposium on VLSI. Massachusetts：IEEE Computer Society Press，2012：392−397.

[18] POREMBA M，ZHANG T，XIE Y. NVMain 2.0：A User-Friendly Memory Simulator to Model(Non-)Volatile Memory Systems. IEEE Computer Architecture Letters，2015，14(2)：140−143.

[19] BEDESCHI F，RESTA C，KHOURI O，et al. An 8 Mb Demonstrator for High-Density 1.8 V Phase-Change Memories[C]//Symposium on Vlsi Circuits. New York：IEEE，2004：99−106.

[20] Micron Technology Inc. Micron 8gb: x4, x8, x16 ddr3l Sdram Description[EB/OL]. [2019−11−08]. https://www.micron.com/products/dram/ddr3−sdram.

（责任编辑：廖友媛）