一种有效的异构盘高能效缓存机制①

窦少彬，杨良怀，龚卫华

(浙江工业大学 计算机科学与技术学院，杭州 310023)

摘 要：固态盘具有低功耗、高性能、耐冲击等优势，硬盘具有高容量、低价格等优势。通过改进文件系统的结构，把固态盘和硬盘结合起来，固态盘作为硬盘的大容量缓存，组成一个我们称之为异构盘的异构系统，其性能接近于固态盘，价格却接近于硬盘。同时，在硬盘有足够空闲时长时，使之关闭以减少能耗。针对大容量缓存，我们采用了合适的树形搜索结构，提出了衰减－增强替换算法获得较高命中率，有效提升存储系统的性能和降低能耗。

关键词：磁盘节能；固态盘；缓存策略；存储系统；能效

Effective Energy-Efficient Buffering Scheme for Hetero-Drive

DOU Shao-Bin, YANG Liang-Huai, GONG Wei-Hua

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, 310023）

Abstract: Compared with traditional hard disk drives (HDDs), solid-state drives (SSDs) are shock-resistant and quieter, and have lower power consumption, lower access time and latency while HDDs have much higher capacity and are cheaper than SSDs. To have the merits of both sides while avoiding their disadvantages, we propose a hetero-drive to combine SSD and HDD by using SSD as a cache for HDD, which creates more idle time for HDD to shut-down for power-saving. At the same time, we put forth a decay-enforcement replacement algorithm (DE algorithm for short) which takes both recency and pastness effect of accesses into account. The experimental results show that our strategy achieves very competitive performance against those traditional buffer replacement algorithms.

Key words: energy conservation; SSD; buffer replacement policy; storage-subsystem; energy-efficiency

在过去的 50 多年里，由于硬盘技术的逐渐成熟，硬盘因其容量大、性价比高等众多优势成为从个人计算机到企业级存储系统等计算机系统的主要存储设备[1]。同时，随着半导体技术的发展和制造工艺的改进，闪存的价格持在持续地下降，性能和容量在不断提升。由 NAND 闪存芯片作为存储介质的固态存储器在某些方面性能已大大超过了传统硬盘。但目前，由于价格和容量的因素，固态盘现在还难以取代硬磁盘在存储器市场上的主导地位。因受到物理因素的限制，磁盘性能受到很大的制约。而盘片的旋转速度和磁头寻道时间已经接近于现有物理技术的极限。而固态盘是由多片闪存芯片以及其他控制芯片组成，不含机械结构，所以突发响应时间非常短。这种多片的体系结构还可以发挥并行存取的优势，获得极高的数据传输速率。

鉴于硬盘的高容量、低价格，固态盘的低容量、高价格和高性能的特征，如果把两者结合起来使用，形成优势互补，就可以组成一个容量和价格接近于硬盘，性能接近于固态盘的存储系统。本文通过改进文件系统的结构，把固态盘和硬盘结合起来，组成一个我们称之为异构盘存储系统。在异构盘系统中，磁盘作为系统的主要存储器，而固态盘则作为磁盘的大容量缓存。由于访问一般都具有时空局部性，因此可以把“热数据”缓存到固态盘中，为硬盘创造更多的空

① 基金项目:国家自然科学基金(61070042);浙江省基金(Y1090096)
收稿时间:2011-03-19;收到修改稿时间:2011-04-11

闲时间，使其进入休眠模式以减少能耗，同时提高存储系统的性能。本文提出了“k-根树”和“衰减－增强”（DE）算法解决高效查找和有效淘汰的问题。结合这两种策略，存储子系统的能效性得到很大的提升。

1　相关工作

固态盘和硬盘的结合使用已经有了一些实例。例如，微软在其产品 Windows Vista 操作系统中使用的 ReadyBoost 和 ReadyDrive[2]技术。这种技术可以有效地提升驱动器的性能，同时也大幅地降低了能耗。Turbo Memory 则结合了 ReadyBoost 和 ReadyDrive 两种技术。混合盘[3]是把闪存集成到硬盘内部，加速系统启动，延长硬盘空闲时间。Kim[4]等提出了在移动设备中把硬盘和固态盘结合起来降低能耗的方案。

综上所述，以上提到的技术可以分成两大类：混合盘[5]方式和组合盘(Combo Drive)[6]方式。它们主要的区别是：混合盘把闪存芯片或固态盘作为硬盘的缓存，混合盘是从硬盘内部的实现。本文提出的异构盘，在实现机制上不同于上述的两种方式。它在不增加硬件的条件下，通过改变存储系统的软件体系结构，隐藏了固态盘的独立存在，从而为用户提供透明访问，同时提高系统的能效。

2　异构盘节能机制

本节介绍异构盘存储系统的体系结构，讲述基于此结构的节能机制。

2.1 异构盘存储系统结构

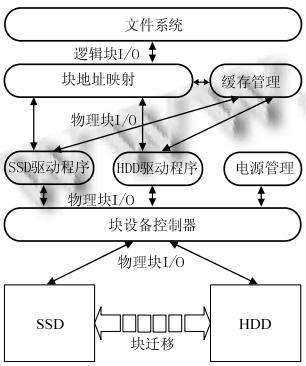

图 1　异构盘系统结构

本文对文件系统进行改造，在硬盘和固态盘的驱动程序的上面加入“块地址映射层”，向上隐藏固态盘的存在。异构盘在文件系统中主要增加三个模块：映射管理模块、缓存管理模块和电源管理模块，见图 1。下面分别就系统的各模块工作机制展开论述。

2.2 映射管理

映射管理的主要功能是把硬盘和固态盘映射为单个的驱动器，使上层文件系统仍可按原有的方式访问存储器。本文通过在固态盘和硬盘的驱动程序上层加入映射管理层，对块 I/O 地址重定向，实现位置透明性。

映射管理主要是通过内存中的映射表来完成，所有缓存在固态盘中的块在映射表中都对应一个表项{B，C，M，　R}。其中，B 为硬盘块号，C 是块 B 缓存在固态盘中的块号，称为“缓存块号”。如果访问块 B，首先查询此映射表。若命中，则直接根据对应的 C 值从固态盘取得数据，否则按 B 访问硬盘。M 为数据块的修改标志。在发生固态盘页面淘汰时，我们为每个数据块维护一个值 R，它表示一数据块被保留权重，R 越大，此块被淘汰的可能性就越小。因此，我们称之为“保留因子”。当需要淘汰时，淘汰 R 值最小的块。由于需要频繁进行数据块查找操作，查找的效率对异构盘系统的性能有较大的影响。为此，本文采用建立多棵树（k-根树）的方法。

定义：k-根树是一颗具有 K（K≥0）个根的树。树中每个结点含有 N（N≥2）个关键字，所有结点的第 1 个关键字（值唯一）构成一颗平衡二叉树，称为分树 T1，所有结点的第 k（1<k≤N）个关键字构成类平衡二叉树，称为分树 T_k。在本文的应用场景下，使用具有两个关键字（磁盘块号 B 和保留因子 R）的 2-根树。假设 $B_1<B_2...<B_{15}$, $R_1≤R_2≤\cdots≤R_{15}$，与其相应的一颗 2-根树如图 2 所示。其中块号关键字的集合（B_1，B_2，…，B_{15}）构成分树 T_b，保留因子关键字的集合（R_1，R_2，…，R_{15}）集合构成分树 T_r。

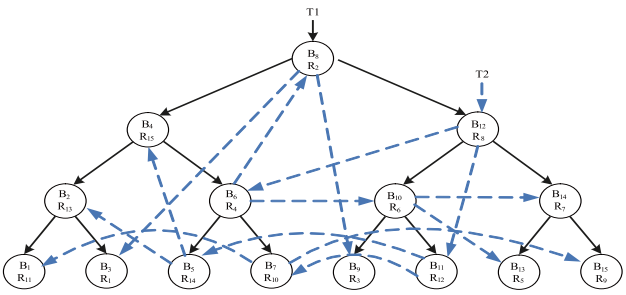

图 2　2-根树

当文件系统对存储系统发送一个块 I/O 请求，首

先从 T_b 的根结点开始搜索，若找到（命中）,则根据对应的 C 值从固态盘中存取数据，否则进入相应的子树继续搜索，这一过程一直重复至命中或相应的子树为空（未命中）。这个过程的时间复杂度为 O(log N)。

当命中时，根据对应的 C 的值从固态盘中存取数据并刷新保留因子 R，此时可能需要调整 T_r，使其仍维持平衡二叉树的性质；当未命中，则按 B 的值对访问硬盘，同时把该块缓存进固态盘，这需把一个新的结点插入此 2-根树。若固态盘已满，应选择 R 最小的块淘汰掉。根据平衡二叉树的性质，T_r 最“左下”的结点即是目标结点。寻找最小 R 的搜索深度不超过 Tr 的高度 H_n，比较次数为 0。因此，使用 2-根树组织映射表，可以有效地保证映射表查找和淘汰的性能。

2.3 缓存管理

本文把固态盘大容量缓存以提高性能和降低能耗。在这种大容量缓存条件下，必须使用高效的缓存算法，才能保证存储系统的性能。由于磁盘块的访问粒度小时间短，算法必须能够非常快地作出决策，否则会降低存储系统的性能。

由于固态盘的容量很大，针对小容量缓存的 FIFO、LFU、LRU、MRU 等淘汰算法并不适合。因此，针对这种应用场景，本文提出了衰减－增强(Decay-Enforce, DE)缓存算法。

DE 算法同时考虑了访问次数和时间两个因素，其主要思想是加强当前访问重要性，对历史访问作衰减，时间越久的访问，其衰减强度也相应加大。本文用“保留因子”表示该数据块重要性。对于数据块 Bi，其对应保留因子为 R_i，初始值为 0。当访问该块时，查找该块是否已经在缓存中。若命中，则更新其保留因子的值 R_i：$R_i \leftarrow R_i \cdot D + E$ （0<D<1）；若未命中,读取硬盘中磁盘块，置 $R_i \leftarrow E$，并缓存，若缓存满，则需要选取最小 R 块进行淘汰。

设固态盘中块的集合为 $\{B_1,B_2,\cdots,B_n\}$，其保留因子集合为 $\{R_1,R_2,\cdots,R_n\}$。如果对块 B_i 进行存取，则对其保留因子进行更新：$R_i \leftarrow R_i \cdot D + E$。记 R_i 更新第 k 次后的值为 $R_i(k)$，已知 $R_i(0)=E$，则：

$R_i(1)= R_i(0)\cdot D+E = E\cdot D+E$，

$R_i(2)= R_i(1)\cdot D+E =(E\cdot D+E)\cdot D+E=E\cdot D^2+E\cdot D+E$，

$R_i(k)=R_i(k-1)\cdot D+E= E\cdot D^k+E\cdot D^{(k-1)}+\cdots+E\cdot D+E$

可知，若块 B_i 访问越频繁，则其保留因子 R_i 的值也就越大。随着 k 的增大，块 B_i 的保留因子 R_i 的极限为

$$\lim_{k\to\infty} R_{i(k)} = \lim_{n\to\infty} E \cdot \frac{\left(1-D^k\right)}{1-D} = \frac{E}{1-D} \quad (*)$$

E 应该根据 D 取适当的值，以保证 R 不会超过一定的范围。假设表示 R 的数据类型有 n 位，则 R 的最大值为 Max(R)=2n-1，则只要 D 和 E 之间的关系满足 E≤ Max(R)· (1-D)，就可保证 R 的值（*）不会溢出。随着运行时间的增加，所有的保留因子最终都可能会趋于同一极限，而使算法失效。为解决这个问题，我们使用计数器 CYC 进行访问计数，当 CYC 达到了某个上限值，则 CYC 清零，对所有快的 R 进行一次衰减，以防趋同。CYC 应该合理取值，若 CYC 取值太大，可能在一个周期内所有的保留因子趋于相同而使得算法失效，若 CYC 太小，则会因更新频繁而开销太大。D 的值决定历史因素在计算当前保留因子中的比重，如果访问模式的周期比较长，D 的值应该取大一些，反之则小一些。

2.4 电源管理

当缓存达到一定的容量，频繁访问的数据块被缓存到固态盘中，给硬盘创造了空闲时间。只要这些空闲状态时间足够长，电源管理使其进入低功耗状态，以减少能耗。在磁盘块的小粒度上，要求管理策略简洁高效，算法能够在极短的时间内完成，因此，本文使用固定超时算法，即当硬盘空闲时间超过一定阀值，就让其关闭，当下一个访问到来时，重新开启进行数据的存取。

3 实验

本文使用 Intel X25-M 和 IBM36Z15 分别作为固态盘模型和硬盘模型进行模拟实验。实验模拟了文件系统，分别使用了两组磁盘块 I/O 追踪[7]（I/O traces）来测试异构盘的能耗和性能。两组追踪的块分布特征如图 3 所示。本文所提的算法以块为调度单位，因所有块的大小相同（本文使用 512B 的块大小），所以在同一硬盘关闭时间阀下，命中率高意味着节能效果好。

不同缓存大小的命中率进行测试结果如图 4 所示。在此两种 OLTP 负载下，当缓存容量达 300MB 时，算法已经达到最大命中率，分别约为 89.3%和 89.4%。上述两种负载的平均传输速率如图 5 所示，它不仅随着缓存大小的变化而变化，而且与负载类型有密切的

关系。实验表明，当缓存大于 300MB 时，其读写速率均已接近于固态盘。关闭时间阀对能耗和硬盘的寿命有很大的影响。因此，要选择一个较合理的值。实验中，在 300MB 缓存条件下，不同时间阈值下的关闭次数和能耗的结果如图 6、图 7 所示。

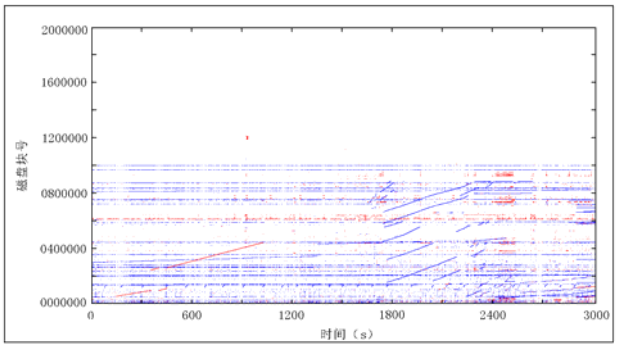

图 3　OLTP 访问分布特征

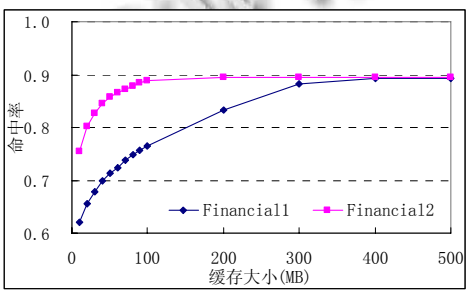

图 4　OLTP 负载下不同缓存的命中率

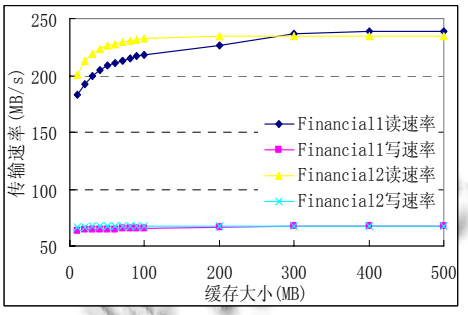

图 5　OLTP 负载下不同缓存大小平均传输速率

结果表明，300MB 缓存下，两种负载分别在 10 秒和 11 秒关闭时间阈值下取得能耗的最低值，关闭次数分别为 10 次和 7 次。因此，在此应用场景下，关闭时间阈值应大于 10 秒。

本文所提的算法以块为调度单位，因所有数据块的尺寸相同，所以在同一关闭时间阈值下，命中率高则意味着节能效果和性能更好。为此，我们进行了 DE 算法与 LRU、LFU，PLRU、CLOCK 以及 FIFO 等替换算法的命中率比较。结果显示，DE 算法可以进行较为准确的预测，取得较高的命中率。与其它算法相比，适应不同的负载和负载变化的能力也较强，表现更为健壮。

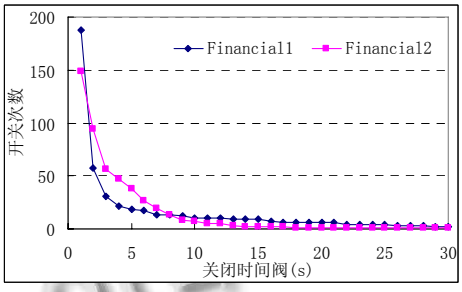

图 6　300MB 缓存下不同关闭时间阀的开/关次数

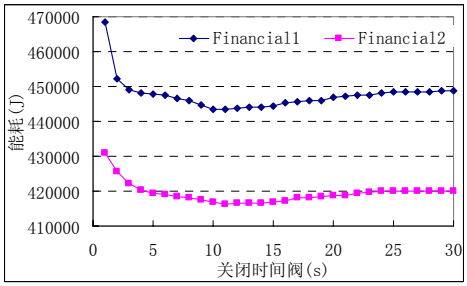

图 7　300MB 缓存下不同关闭时阀的能耗

4　结论

本文提出的异构盘方案把固态盘和硬盘结合起来，组成一个容量可以和硬盘相比较，性能可以和固态盘相比较的混合存储系统，该方案对存储子系统的软件体系结构进行了改进，把固态盘作为硬盘的大容量缓存，采用 DE 淘汰算法以磁盘块的粒度进行缓存操作。同时，当硬盘空闲状态超过一定的时长，让其进入低功耗模式以减少能耗。实验表明，异构盘方案不仅大幅地提升了存储子系统的性能，还能有效降低系统能耗。在实验负载环境下，读、写速率分别平均约提高了 265%和 25%，能耗平均减少了 28%左右。

参考文献

1 Sanvido MA, Chu FR. Nand flash memory and its role in storage architectures. Proc. of the IEEE, 2008,96(11):1864–1874.

2 Panabaker R. Hybrid hard disk and readydrive technology: Improving performance and power for windows vista mobile

(下转第 106 页)

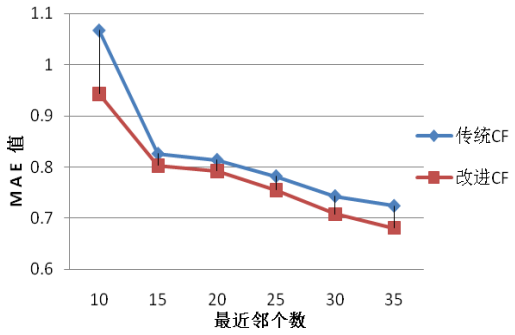

图 2　由测试集 1 得到的 MAE 值对比

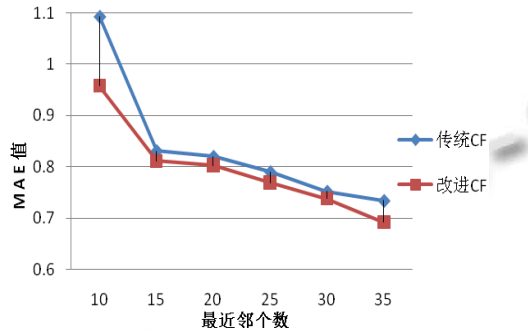

图 3　由测试集 2 得到的 MAE 值对比

4　结论

与传统协同过滤推荐算法比较，本文提出的改进协同过滤推荐算法最大不同的是：从评分数据稀疏问题和产生最近邻的相似性计算出发，讨论了 I-U 评分矩阵的修正问题和仅依靠用户打分是否能成为最近邻的问题，并提出了改进的方法，以用户行为对应分值替代的方式修正了 I-U 评分矩阵并以权重系数 K 约束最近邻用户的计算，使得推荐算法能更好的产生最邻近用户集。实验结果表明，本算法较显著地提高了推荐质量。

参考文献

1 王茜,杨莉云,杨德礼.面向用户偏好的属性值评分分布协同过滤算法.系统工程学报,2010,(4):561−568.

2 黄晓斌.基于协同过滤的数字图书馆推荐系统研究.大学图书馆学报,2006,(1):53−57

3 Kin BM, Li Q, Park CS, et al. A new approach for combining content-based and collaborative filters. Journalof Intelligent Information Systems, 2006,27:79−91.

4 周敏,周继鹏,丁光华.PSL:针对大规模数据应用的并行 Slope One 算法.科学技术与工程,2010,3:711−714.

5 Tian W, Xu J, Pend YQ. CF Improvement Based on Probabilistic Analysis of Discrete Explicit Rating Vector. Proc. of the 2009 First IEEE International Conference on Information Science and Engineering, 2009,9:814−816.

6 Sarwar B, Karypis G, Konstan J, et al. Item-Based Collaborative Filtering Recommendation Algorithms. Proc of the 10th Int'l World Wide Web Conf, 2001,285−295.

7 Breese J, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proc. of the 14th Conference on Uncertainty in Artificial Intelligent. 1998, 43−52.

8 Herlocker JL, Konstan JA, Terveen LG. Evaluating Collaborative Filtering Recommendation System. ACM Trans. on Information System, 2004,22(1):5−53.

9 冯克鹏.基于协同过滤的数字图书馆推荐系统研究.软件导刊,2010,5:16−18.

(上接第 102 页)

pcs. http://www. microsoft. com/whdc /winhec/pres06.mspx. 2006.

3 Panabaker R. Hybrid Hard Disk & ReadyDriveTM technology: Improving performance and power for Windows Vista Mobile PCs. WinHEC, 2006.

4 Kim YJ, Kwon KT, Kim J. Energy-effcient fle placement techniques for heterogeneous mobile storage systems. EMSOFT Conference, 2006.

5 Microsoft. ReadyDrive and hybrid disk. http://www. microsoft.com/whdc/system/sysperf/perfaccel.mspx. 2010.

6 Payer H, Sanvido MAA. Combo Drive: Optimizing Cost and Performance in a Heterogeneous Storage Device. Workshop on Integrating Solid-state Memory into the Storage Hierarchy, 2009.

7 http://traces.cs.umass.edu/index.php/Storage/Storage. 2007.