# 组合盘中结合访问次数与能量代价的文件缓存替换策略①

周 健, 杨良怀, 龚卫华

(浙江工业大学 计算机科学与技术学院，浙江 杭州 310014)

**摘 要**：如何有效地降低存储子系统能耗是近几年研究的热点议题。新型非易失、抗震、低功耗闪存及固态盘的出现给存储子系统节能带来了新的机会。但其每单位价格昂贵，目前难以替代硬盘的角色。结合硬盘和固态盘的优势，本文采用组合盘（由硬盘和固态盘组成）节能。结合文件访问次数和能量代价，我们提出了改进的文件缓存替换策略 FEBR(Frequency & Energy-based replacement)。实验采用两个真实办公用户数据，结果表明组合盘方案是可行的，节能百分比可达 70%～80%；与经典替换算法、最新较好的 ARC 算法以及理想最优页面 OPT 算法进行了详细比较，FEBR 优于其它策略。

**关键词**：组合磁盘；节能；能效；缓存替换算法

## Frequency and Energy-Based Replacement Scheme for Heterogeneous Drive

ZHOU Jian, YANG Liang-Huai, GONG Wei-Hua

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014)

**Abstract**：How to reduce the energy consumption of storage sub-system effectively has gained a lot of attention these years. The emergence of non-volatile, shock resistant and low power flash memory and solid state driver (SSD) brings a new opportunity for power-saving in storage sub-system. However, it still cannot totally replace hard disk for its higher cost per storage unit. This paper focuses on the heterogeneous drive scheme (heter-Drive for short) for both energy-conservation. By considering both file access frequencies and energy cost, we propose an improved file cache replacement scheme called FEBR (Frequency and Energy-based Replacement). We collected four real-world office users' file access data for experiments. The results show that heter-Drive does works well and saves as high as 70% ~ 80% of energy; the extensive comparisons with the classic replacement algorithms, the widely used ARC and the optimal algorithm OPT show that FEBR is consistently better than other alternatives.

**Key words**：heterogeneous drive; energy conservation; energy efficiency; cache replacement policy

## 1 引言

存储子系统的能耗在通用计算机系统总能耗中占很大比例，约 20-30%[1,2]。随着新型非易失、抗震、低功耗闪存的出现，NAND 闪存以及闪存固态盘的广泛使用，为降低存储子系统能耗带来了新的机会。表 1、表 2 所示为 3 个硬盘[3]和 1 个固态盘[4]的具体参数。从表中观察到固态盘顺序读、写的速度分别为硬盘的 2.7～10 倍,0.75～2.8 倍，功率比硬盘低了 1～2 个数量级，性能优、功耗低。但 Narayanan 等[5]认为固态盘在目前的成本/GB 状态下高出硬盘 3 到 3000 倍，还难以替代硬盘的角色。

因此，结合固态盘功耗低、读取速度快的优势，硬盘价格低、容量大的特点，我们采用了类似[6]中讨论的组合盘方案，系统结构如图 1 所示。该方案在已有的存储系统增加固态盘作为文件缓存，并在文件系统底层增加文件缓存管理器，实时监控系统的文件操作，检测用户的访问模式；文件缓存替换机制利用文件缓存管理器检测到的信息对固态盘中的文件按重要

① 基金项目:国家自然科学基金(61070042)以及浙江省自然科学基金(Y1090096)
收稿时间:2011-09-16;收到修改稿时间:2011-10-29

程度排序，依此决定当固态盘空间满时优先选择淘汰哪些文件，以便合理地把文件分配至固态盘和硬盘中。如此便将“热点”的工作集数据缓存至固态盘中，大部分数据请求可由固态盘独立完成，从而可以延长磁盘的待机时间。这样，组合盘存储系统的容量、价格和硬盘相当，能耗、性能却接近固态盘，在这四个方面均达到了平衡。

本文对组合盘节能缓存机制提出了基于频率和能量替换算法 FEBR。通过收集多个真实用户数据，与经典替换算法、最新的 ARC 算法以及理想最优页面替换算法 OPT 进行了详细比较，实验结果表明 FEBR 优于其余方案。

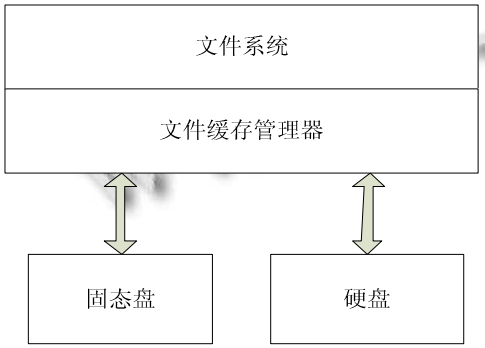


图 1 改进的文件系统结构

表 1 Intel X25-M SATA SSD 参数

| | |
|---|---|
| Power (Active) | 0.150 W |
| Power (Idle) | 0.060 W |
| Seq Read | 250 MB/s |
| Seq Write | 70 MB/s |
| 4k rand read | 35000 IOPS |
| 4k rand write | 3300 IOPS |
| R/W latency | 65 μs |

文章其余部分组织如下：第二节详细介绍组合盘中基于频率和能耗代价文件缓存替换策略 FEBR；第三节进行实验评估；最后给出了总结。

## 2 频率和能耗代价结合的文件缓存替换策略

本文与效果最好、最新或最经典的几个替换算法进行比较实验，探索在办公环境台式机组合盘中这些策略能效和性能问题：LRU[7]、LFU[8]、MRU、ARC[9]、FBR[10]、OPT[7]。Liu 等[6]初步实验表明 FBR 节能效果优于 LRU。但基于文件的缓存替换场景，当缓存满时，原始 FBR 算法优先选择淘汰旧区最小访问计数的文件而并没有考虑每个文件大小的不同以及淘汰每个文件的能量代价。结合上述因素的考虑，基于 FBR 算法，我们提出了一个改进的 FEBR 算法，详见如下。

表 2 3 个不同转速的硬盘参数

| Disk Type / Parameters | | IBM 36Z15 | Western Digital WD2500JD | IBM 40GNX 40GNX |
|---|---|---|---|---|
| RPM | | 15,000 | 7,200 | 5,400 |
| Average seek time(ms) | | 3.4 | 8.9 | 12 |
| Average rotational latency (ms) | | 2 | 4.2 | 5.5 |
| IDR(MB/sec) | | 55 | 93.5(max) | 25 |
| Power(Watt) | Active | 13.5 | 13.25(Seek) 10.6(R/W) | 3.0 |
| | Idle | 10.2 | 10.0 | 0.82 |
| | Standby | 2.5 | 1.8 | 0.25 |
| Energy(Joule) | Spin Down | 13.0 | 6.4 | 0.4 |
| | Spin Up | 135.0 | 148.5 | 8.7 |
| Time(Sec) | Spin Down | 1.5 | 4.0 | 0.5 |
| | Spin Up | 10.9 | 9.0 | 3.5 |

对于使用访问次数（频率）的缓存算法，需要维护频率信息。本文所提文件缓存管理器维护了固态盘中文件的文件名 $F_i$、修改标志 $m_i$(修改为 1，未修改为 0)、文件大小 $s_i$、访问次数 $f_i$（文件 Fi 的访问次数），如表 3 所示。

表 3 文件缓存管理器维护的数据

| 文件 $F_i$ | $F_1$ | $F_2$ | … | $F_i$ | … |
|---|---|---|---|---|---|
| 修改标志 $m_i$ | *0/1* | *0/1* | … | 0/1 | … |
| 文件大小 $s_i$ | $s_1$ | $s_2$ | … | $s_i$ | … |
| 访问次数 $f_i$ | $f_1$ | $f_2$ | … | $f_i$ | … |

算法中涉及的固态盘、硬盘有关参数定义如下。硬盘有三种工作状态：活动状态（active）、空闲状态(idle)和待机状态(stdby)。磁盘可以从空闲状态进入待机状态，此过程称为旋转减速(spin down)，其能耗记为 $E_{idle\to stdby}$，其时延为 $\tau_{idle\to stdby}$；硬盘可以从待机状态

回到活动状态，此过程称为旋转提速(spin up)，其能耗记为 $E_{stdby\text{-}act}$，其时延为 $\tau_{stdby\text{-}act}$。其余用到的几个参数定义如下：

$p_i$：访问 Fi 的概率，取 $p_i=f_i/f_{max}$，$f_{max}$ 是固态盘文件集合中最大文件访问次数；

$p_{spin\text{-}up}$：硬盘处于待机状态的概率；

$R^r_{SSD}$：固态盘的读取速率；

$R^w_{SSD}$：固态盘的写入速率；

$R_{HD}$：硬盘的传输速率；

$PWR^r_{SSD}$：固态盘读操作功率；

$PWR^w_{SSD}$：固态盘写操作功率；

$PWR_{SSD}$：指 $PWR^r_{SSD}$ 或者 $PWR^w_{SSD}$，两者之一。

$PWR_{HD}$：硬盘的功率；

H：文件访问序列的整体命中率。

本文磁盘电源管理策略采用固定超时算法[11]。在实验中统计了不同超时阈值下带来的能量、时延、硬盘启停次数等信息。

缓存中的不同文件淘汰到硬盘而后再进行访问的代价是不同的。对于文件 $F_i$，其大小 $s_i$，访问概率 $p_i$，修改标记 mi，则其淘汰到硬盘的代价 $C_i$ 为写到硬盘的代价与将来访问该文件的代价之和：

$$C_i = m_i\left(\frac{s_i}{R^r_{SSD}}\cdot PWR^r_{SSD}+\frac{s_i}{R_{HD}}\cdot PWR_{HD}\right)$$
$$+p_i\cdot\left[\frac{s_i}{R_{HD}}\cdot PWR_{HD}+p_{spin-up}\cdot E_{stdby-act}\right]$$

缓存命中率的高低和磁盘进入待机态可能性是成正比的，为方便计，令 $p_{spin\text{-}up}\approx H$。假设这些文件访问是独立的，则其总代价为：$C=\sum Ci$。对于命中率较高的情形，硬盘处于待机状态的概率也大。硬盘从待机态进入活动态要消耗较多能量 $E_{stdby\text{-}act}$。对于概率相同(设为 p)的较为久远的文件而言，优先淘汰较大文件比之于淘汰多个较小文件的代价要小。如在读访问居多模式，淘汰一个大文件 50M 比淘汰 10 个 5M 文件的期望代价要小。因此，结合代价模型改进 FBR 的替换策略是值得尝试的，我们称之为 FEBR(Frequency & Energy-Based Replacement)：FEBR-0 优先淘汰旧区中能量代价 C 最大的文件；FEBR-1 优先淘汰旧区中访问频率为 1 且能量代价 C 最大的文件。FEBR 与 FBR 的不同之处在于，FBR 仅考虑频率(淘汰旧区中访问频率最小的文件)，FEBR 则同时考虑了频率和能量代价。

## 3 实验评价

### 3.1 实验数据收集

为了获得办公环境 Windows 用户对文件使用的方式，我们对开源软件 FileMon 进行了改写，记录文件打开、读取、写入、新增、删除等操作。共收集了 2 个用户的数据，汇总信息见表 4。

表 4　Trace 基本信息

| 追踪 | 总访问条目数 | 天数 | 平均文件大小 |
|---|---|---|---|
| Trace 1 | 170197 | 33 | 0.235MB |
| Trace 2 | 96578 | 37 | 4.497MB |
| Trace 3 | 73204 | 34 | 3.310MB |
| Trace 4 | 57468 | 22 | 0.593MB |

### 3.2 实验方法

本文对收集的数据进行了仿真实验，通过回放收集的文件访问追踪(Trace)来评价所有参与比较的缓存算法的性能。实验中所采用固态盘、硬盘参数见表 1、表 2。本文所设的参数如下：

(1)FBR：新、中、老三个分区所占缓存百分比分别为 30%，50%，20%。

(2)FEBR-0、FEBR-1：参数与 FBR 相同。

(3)ARC：P 的增减量为最新访问文件的大小。历史数据 B1+B2 不大于缓存大小。

实验采用的评价指标有：文件访问命中率、节能效率、文件存取平均响应时间。

### 3.3 实验比较

(1) 命中率

以 WD2500JD(7200RPM/sec)硬盘进行实验，各个算法在不同缓存大小下命中率如图 2 所示。整体上来看，OPT、FEBR-0、FEBR-1、FBR 是所列算法中命中率一直领先的。OPT 算法在缓存较小时，命中率并非最高，已不是最优算法。当缓存增大时，OPT 逐渐符合理想状态，命中率优于其他算法。FEBR-0、FEBR-1 算法大部分情形下高出 FBR，其中 FEBR-0 有时略低于 FBR(如 TRACE1 中，缓存大小 512MB 时)，表现不稳定；而 FEBR-1 保持稳定，一直高于 FBR。这是因为 FEBR-0 优先选择淘汰的是旧区中能量代价最大的文件并未充分考虑文件的访问频率；而 FEBR-1 避免了 FEBR-0 的缺陷。

(2) 能耗

观察在不同的硬盘的超时阈值，分别为 10s，60s，

180s；不同缓存大小，分别为 512MB，1536MB 时，各个 TRACE 的能耗情况。非组合盘的能耗是（单位：104J）：Trace1=1426，Trace2=1227。我们以三个硬盘实验，由于篇幅所限，只呈现 WD2500JD（7200RPM/sec）实验结果，如图 3 所示。整体上看，TRACE1～2 的节能效果良好，可达 70%～80%。结合图 2、图 3 来分析，命中率与节能百分比是成正比的，FEBR-0、FEBR-1、FBR 保持了良好的节能效果。缓存大小增大节能效果越好；硬盘超时阈值越小，节能效果越好。缓存越小、硬盘超时阈值越大时，各个算法节能效果差异越明显。

| SSD | OPT | FEBR-0 | FEBR-1 | FBR | ARC | LRU | LFU | MRU |
|---|---|---|---|---|---|---|---|---|
| 512MB | 85.96 | 86.28 | 86.92 | 86.79 | 82.49 | 84.69 | 81.65 | 63.84 |
| 1024MB | 86.99 | 86.90 | 86.99 | 86.88 | 85.51 | 85.54 | 82.27 | 70.28 |
| 1536MB | 87.14 | 86.99 | 87.06 | 86.93 | 86.60 | 85.98 | 82.27 | 71.05 |
| 2048MB | 87.14 | 87.00 | 87.04 | 86.96 | 85.74 | 86.13 | 81.82 | 68.99 |

(a) Trace 1

| SSD | OPT | FEBR-0 | FEBR-1 | FBR | ARC | LRU | LFU | MRU |
|---|---|---|---|---|---|---|---|---|
| 512MB | 89.50 | 93.04 | 92.75 | 92.48 | 85.18 | 84.10 | 76.96 | 66.75 |
| 1024MB | 93.34 | 94.12 | 93.02 | 92.68 | 89.21 | 89.18 | 84.36 | 69.46 |
| 1536MB | 94.28 | 94.25 | 93.88 | 93.24 | 92.27 | 92.37 | 86.26 | 76.45 |
| 2048MB | 94.72 | 93.93 | 93.87 | 93.01 | 92.67 | 92.96 | 89.64 | 77.73 |

(b)Trace 2

图 2 WD2500JD 硬盘两个 Trace 的命中率

(3) 文件请求平均响应时间

据三个硬盘参数进行实验，由于篇幅所限，只呈现 IBM 40GNX(5400 RPM/sec)实验结果，如图 4 所示。其中 Trace1、Trace2 单硬盘的文件请求平均响应时间分别为 27ms，197ms。组合盘平均请求访问时间普遍低于单硬盘访问时间，较好时为后者的 1/6。这是因为 IBM 40GNX 硬盘的传输速度是 WD2500JD 的 1/4，读写速度分别为固态盘的 1/10、1/3，固态盘与硬盘传输速度的进一步扩大弥补了硬盘启停造成的时延。整体上看，OPT、FEBR-0、FEBR-1、FBR 优于其他算法。

## 4 总结

本文采用组合盘作为降低能耗的存储设备，探索在办公环境下基于文件粒度的固态盘缓存替换机制的设计，比较了各种经典算法、文献中认为较好替换算法，在命中率、能效、文件请求平均响应时间等方面汇报了结果。实验结果表明组合盘节能效果明显，可达 70%～80%。在 FBR 算法基础上结合能耗代价计算公式，我们提出了基于文件访问频率与文件存取能量代价的 FEBR 算法，结果表明能耗代价模型对文件的淘汰选择具有一定的指导意义。

(a)Trace 1 512MB

| $10^4$J | 10S | 60S | 180S |
|---|---|---|---|
| OPT | 315 | 354 | 421 |
| FEBR-0 | 316 | 355 | 423 |
| FEBR-1 | 314 | 352 | 418 |
| FBR | 315 | 353 | 420 |
| ARC | 325 | 370 | 444 |
| LRU | 322 | 364 | 436 |
| LFU | 325 | 368 | 441 |
| MRU | 435 | 482 | 560 |

(b)Trace 1 1536MB

| $10^4$J | 10S | 60S | 180S |
|---|---|---|---|
| OPT | 312 | 349 | 414 |
| FEBR-0 | 313 | 351 | 417 |
| FEBR-1 | 313 | 350 | 416 |
| FBR | 313 | 351 | 417 |
| ARC | 316 | 355 | 423 |
| LRU | 316 | 356 | 423 |
| LFU | 319 | 360 | 431 |
| MRU | 429 | 473 | 547 |

(c)Trace 2 512MB

| $10^4$J | 10S | 60S | 180S |
|---|---|---|---|
| OPT | 240 | 254 | 283 |
| FEBR-0 | 239 | 250 | 276 |
| FEBR-1 | 239 | 251 | 278 |
| FBR | 239 | 252 | 278 |
| ARC | 244 | 261 | 296 |
| LRU | 245 | 262 | 298 |
| LFU | 252 | 276 | 316 |
| MRU | 354 | 394 | 458 |

(d)Trace 2 1536MB

| $10^4$J | 10S | 60S | 180S |
|---|---|---|---|
| OPT | 238 | 249 | 273 |
| FEBR-0 | 238 | 249 | 273 |
| FEBR-1 | 238 | 249 | 274 |
| FBR | 238 | 249 | 274 |
| ARC | 239 | 251 | 277 |
| LRU | 239 | 251 | 277 |
| LFU | 243 | 258 | 291 |
| MRU | 346 | 377 | 429 |

图 3 WD2500JD 硬盘不同参数下的能耗比较

| (ms) | 10S | 60S | 180S |
|---|---|---|---|
| OPT | 49 | 28 | 19 |
| FEBR-0 | 50 | 29 | 19 |
| FEBR-1 | 49 | 28 | 19 |
| FBR | 47 | 26 | 17 |
| ARC | 54 | 31 | 21 |
| LRU | 53 | 31 | 20 |
| LFU | 57 | 33 | 22 |
| MRU | 71 | 39 | 27 |

(a)Trace 1, 1G

| (ms) | 10S | 60S | 180S |
|---|---|---|---|
| OPT | 38 | 33 | 31 |
| FEBR-0 | 84 | 78 | 76 |
| FEBR-1 | 84 | 79 | 77 |
| FBR | 83 | 78 | 76 |
| ARC | 43 | 37 | 34 |
| LRU | 43 | 37 | 34 |
| LFU | 50 | 41 | 37 |
| MRU | 175 | 149 | 133 |

(b)Trace 2, 1G

图 4 IBM40GNX 不同参数下的平均响应时间比较

理想的基于页面最优算法 OPT 在基于文件粒度的

(下转第 207 页)

在实验中指定 $C_{\min}$ 的值为 60，指定 $C_{\max}$ 的值为 180。图 2 为对 FVC2002 DB3_A 24_4.tif 增强的结果。其中图 2(a)为原始指纹图像，图 2(b)为对(a)Gabor 增强后的结果，图 2(c)为对 2(b)做二值化之后的结果，图 2(d)为对图 2(a)用本文方法增强后的结果，图 2(e)为对 2(d)做二值化之后的结果。通过图 2 可以看出本文的增强算法对于有很大噪声的图像的增强效果要明显好于 Gabor 增强。之所以有这样的结果一方面是因为相位是在考虑局部邻域图像基础上得到的，具有一定的鲁棒性；另一方面是因为对相位又进行了一次滤波。

## 4 结论

本文提出了灰度级相位的概念，并实现了一个根据灰度级相位增强指纹图像的方法。该方法首先计算指纹区域内每个像素点的灰度级相位，然后对灰度级相位进行滤波，最后根据滤波后的灰度级相位构建指纹图像。该方法在增强图像时实际上相当于进行了两次滤波：一次是通过计算灰度级相位过滤掉一些噪声的影响；另一次是对灰度级相位滤波进一步减少噪声。对 FVC 指纹库的测试结果也表明该方法对低质量的指纹图像增强效果好。

## 参考文献


1 Hong L, Wan YF, Jain AK. Fingerprint image enhancement: algorithm and performance evaluation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1998,20(8): 777–789.

2 Ko T. Fingerprint enhancement by spectral analysis techniques. In Proceedings of the Applied Imagery Pattern Recognition Workshop, Washington DC, USA, 2002:133–139.

3 王科俊,李雪峰,赵钥.基于方向滤波的指纹图像增强算法研究.模式识别与仿真,2009,28(7):54–56.

4 卞维新,徐德琴.基于自适应方向滤波器的指纹图像增强.微电子学与计算机,2009,26(3):185–188.

5 王莹,苏成利.指纹图像增强方法研究.科学技术与工程,2010,10(1):94–98.

6 Luo X, Tian J. Knowledge Based Fingerprint Image Enhancement. International Conference on Pattern Recognition. Barcelona, Spain, 2000,4:783–786.

7 Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK. FVC 2002:Second Fingerprint Verification Competition. In Proc. of the International Conference on Pattern Recognition. Quebec, Canada, 2002,3:811–814.


(上接第 211 页)

缓存机制中已不是最优。进一步工作中，将对更多的替换算法进行对比实验。同时，注意到本文研究中结合能量代价计算的缓存替换策略在本场景发挥了积极的作用，将在这方面作更深入的探索。

## 参考文献


1 Douglis F, Krishnan P, Marsh B. Thwarting the power hungry disk. USENIX Winter Conference. 1994,292–306.

2 Greenawalt P. Modeling power management for hard disks. Workshop on Modeling, Analysis, and Simulation on Computer and Telecommunication Systems, 1994,62–66.

3 Deng Y. What is the future of disk drives, death or rebirth?. ACM Computing Surveys, 2011,43(3).

4 Intel X18-M/X25-M SATA Solid State Drive Product Manual. 2009, May. http://download.intel.com/design/flash/nand/mainstream/mainstream-sata-ssd-datasheet.pdf.

5 Narayanan D, Thereska E, et.al. Migrating server storage to SSDs: analysis of tradeoffs. 2009, EuroSys.

6 Liu S, Cheng X, Guan X, Tong D. Energy efficient management scheme for heterogeneous secondary storage system in mobile computers. ACM Symposium on Applied Computing, 2010,251–257.

7 Belady LA. A study of replacement algorithms for virtual storage computers. IBM Systems Journal, 1966,5(2):78–101.

8 Mattson RL, Gecsei J, Slutz DR, Traiger IL. Evaluation Techniques for Storage Hierarchies. IBM Systems Journal, 1970,9(2):78–117.

9 Megiddo N, Modha DS. ARC: A Self-Tuning, Low Overhead Replacement Cache. USENIX Conference on File and Storage Technologies (FAST), 2003,115–130.

10 Robinson JT, Devarakonda MV. Data cache management using frequency-based replacement. ACM SIGMETRICS Conference. 1990.134–142.

11 Li K, Kumpf R, Horton P, Anderson T. A Quantitative Analysis of Disk Drive Power Management in Portable Computers. USENIX Winter Conference. 1994.279–292.