

# 页面置换算法调研报告

ljt12138

2019 年 4 月 4 日

## 1 概述

在页式虚拟存储计算机中，内存被看作是磁盘的缓存，操作系统通过将常用的虚拟内存页放入物理内存页帧中，将不常用虚拟内存页放入磁盘中来提供更大、更自动化的虚拟存储。页面置换算法在上世纪曾是很热门的研究方向，Belady的 [1]中根据使用以往信息的多少对算法大致归类，并给出了离线下的最优算法；理论上，对于页面置换问题竞争性分析的研究有许多成果，Fiat等人的 [2]对于更一般的k-server问题证明了随机标记法的竞争比，并分析了随机算法竞争比的下界。todo

确定性页面置换算法的竞争比不会小于 $k$ ，通过在算法中引入随机因素，可以期望将竞争比降低至 $\log k$ 。可以证明，随机页面置换算法期望竞争比的下界是 $\log k$ 。

## 2 确定性算法

页面置换问题可以被这样描述：给定一台支持 $h$ 个逻辑页面、拥有 $k$ 个页面大小物理内存的机器，选择每次的置换页面使得总的置换次数最少。

页面置换算法可以分为离线算法（off-line）和在线算法（on-line）。离线算法是指整个访存序列已经预先给定，而在线算法是指访存序列是动态给出的。很显然，真实情景下的页面置换算法必须是在线的。

Belady在其 [1]中最早给出了称为MIN算法的最优离线算法。另一种简单的算法FIF（Further-in-future），通过总是换出下一次访问最晚的页面，算法可以做到最小化缺页次数 [5]。

虽然最优算法需要使用未来的信息，但其可以作为分析在线算法的一个标准。通过缺页次数与最优算法缺页次数的比可以简单评价一个在线算法的优劣。设一个算法 $A$ 在访存序列 $S$ 上的缺页次数为 $Miss(A, S)$ ，如果对于任意访存序列 $S$ ，都有：

$$Miss(A, S) \leq f(k, h) \times OPT(S) + O(1)$$

我们可以称，算法 $A$ 是 $f(k, h)$ -Competitive的， $f(k, h)$ 可以称为算法 $A$ 的竞争比。

### 2.1 确定性在线算法竞争比的下界

**定理 2.1** 任何确定性在线算法 $C$ 的竞争比至少是 $k$ 。

**证明 2.1** 考虑仅有 $k + 1$ 个逻辑页面的长度为 $l$ 的访存序列，每次访问的页面恰好是 $C$ 上一次换出的页面，那么每一次访问都会发生缺页；在最优算法中，每次缺页发生之后，下一次发生缺页一定是在 $k$ 次之后（因为换出的是下一次访问最晚的页面），因而总的缺页次数不会超过 $k + l/k$ 。有：

$$Miss(C, S) \geq k \times OPT(S) + O(1)$$

### 2.2 标记算法

标记算法（Marking Algorithm）是一类置换算法的总称。假设每个在内存中的页面都有一个标记位 $A$ ，初始值为0。未标记的页面会在访问时被标记，当所有页面都被标记时，将所有页面的标记重新置为0。标记算法是那些只会将未标记的页面置换出去的算法。

许多根据访问排序的页面置换算法都是标记算法，例如最近未使用算法（NRU）与最近最久未使用算法（LRU）。

**引理 2.1** LRU是标记算法。

**证明 2.2** 考虑使用反证法。如果LRU不是标记算法，换言之在某个时刻其换出了一个未标记的页 $p$ ，并且存在一个标记的页 $q$ 没有被换出。在最后一次执行所有页标记置0后， $q$ 被访问了至少一次， $p$ 没有被访问过，因而 $q$ 上一次访问比 $p$ 要近。根据LRU算法的定义， $p$ 不会被置换出，矛盾。

**定理 2.2** 任何标记算法都是 $k$ -Competitive的。

**证明 2.3** 将序列按顺序划分为 $t$ 个段 $b_1, b_2, \dots, b_t$ ，每个段 $b_i$ 中仅包含至多 $k$ 种不同的页面，且 $b_1$ 尽可能长，在此前提下 $b_2$ 尽可能长，以此类推。即：对于一个以 $s_i$ 开头的段 $b_i$ ， $b_{i+1}$ 的开头 $s_j$ 是最小的 $j$ ，满足 $\{s_i, s_i+1, \dots, s_j\}$ 包含 $k+1$ 个不同的位置。我们称这是访存序列的分割。

用归纳法可以证明，在每一段末尾，标记算法总会使得所有段被标记。那么在下一段开头所有段都被清除标记，且这些未标记的段要么被重新访问，要么被换出，这一段内的换出次数不会超过 $k$ 。那么总的换出次数不超过 $t \times k$ 。

接下来我们说明最优算法至少会发生 $t$ 次缺页。考虑每一个段开头的页面 $p_1, p_2, \dots, p_t$ 。如果 $p_i$ 没有发生缺页，说明 $p_i$ 在内存中。根据划分段的定义，上一次 $p_i$ 被引用一定不在 $b_{i-i}$ 中，因而 $b_{i-1}$ 可以使用的内存页面只有 $k-1$ 个。那么除掉 $p_{i-1}$ 发生的缺页，根据鸽巢原理，还会至少发生一次缺页。所以总的缺页次数至少是 $t$ 。由于：

$$Miss(Marking, S) \leq t \times k \leq k \times OPT(S)$$

任何标记算法都是 $k$ -Competitive的。

一些确定性算法的竞争比结论如下 [7]。

- LRU算法是 $k$ -Competitive的；这是上面结论的平凡推论。
- FIFO, Clock算法是 $k$ -Competitive的；他们都是所谓Conservative algorithms [8]。
- LIFO, LFU算法不是Competitive的，即算法缺页次数和最优解的缺页次数之比可以无限大 [9]。

### 3 随机算法的竞争比

类似的，我们定义随机算法 $R$ 是 $f(h, k)$ -Competitive的，如果：

$$\mathbb{E}[Miss(R, S)] \leq f(h, k) \times OPT(S) + O(1)$$

在线算法应对输入的过程可以看成一个双人零和博弈。算法是“Defender”，为了让代价最小化；输入是“Hacker”，为了让代价最大化。很显然，对于任何一方而言，拥有关于对方策略的信息越多，其期望收益就越大。

### 3.1 随机标记算法

一种有效的随机置换算法被称为随机标记算法 (Random Marking), 下文中我们简称其为RM。其思路类似于一般的标记算法, 但每次需要换出时会从所有未被标记的页面中随机选出一个换出。Fiat首先证明了RM是 $2 \log k$ -Competitive的 [2]。

**定理 3.1** 随机标记算法是 $2 \log k$ -Competitive的。

**证明 3.1** 考虑访存序列的分割 $b_1, b_2, \dots, b_t$ , 考虑一个段 $b_i$ 中的元素。我们称一个页面是**新的**, 如果它在 $b_{i-1}$ 中没有被引用, 在 $b_i$ 中还没有被引用; 称一个页面是**旧的**, 如果它在 $b_{i-1}$ 中被引用过, 在 $b_i$ 中还没有被引用。不妨设 $b_i$ 中对新页面的引用有 $l_i$ 次, 对旧页面的引用有 $k - l_i$ 次。

我们首先证明, 最优算法在 $b_i$ 执行的开销至少是 $\sum l_i/2$ 。假设第 $b_i$ 中对新页面的引用发生了 $p_i$ 次缺页, 那么在 $b_{i-1}$ 中另外的 $l_i - p_i$ 个页面都始终在内存中还没有被换出, 因而在 $b_{i-1}$ 中可用的页面只有 $k - (l_i - p_i)$ 个, 至少发生了 $l_i - p_i$ 次缺页。考虑将这两者相加得到:

$$\sum_i p_i + (l_i - p_i) = \sum_i l_i$$

一次缺页至多会被当前段计算一次, 被下一段计算一次, 因此总的缺页次数不会小于上面结果的一半, 即 $\sum l_i/2$ 。

接下来我们说明RM算法在段 $b_i$ 的缺页次数不会超过 $l_i H_k$ , 从而总的缺页次数不会超过 $H_k \sum l_i$ 。其中 $H_k$ 是调和级数前 $k$ 项的和。对于任何标记算法, 在一个段开始时, 内存中总是存放上一个段的所有内容。因此, 每一个新页面的引用都会导致缺页, 我们只需要关心旧页面产生的缺页次数, 即由于被 $b_i$ 中导致的缺页换出的旧页面。很明显, 最差情况下所有新页面的引用都在旧页面的引用之前。

如果一个页面 $P$ 将页面 $Q$ 换出, 我们连接一条从 $P$ 到 $Q$ 的有向边, 那么只有引用时入度为1的点会发生缺页。设旧页面依次是 $O_0, O_1, \dots, O_{k-l_i+1}$ 。当处理 $O_j$ 时, 发生缺页且可能换出 $O_j$ 的, 就是那些入度为1, 出度为0的点, 而这样的点恰好有 $l_i$ 个。这 $l_i$ 个点会在还未被换出的 $k - j$ 个旧页面中等概率取出恰好 $l_i$ 个, 那么 $O_j$ 被取出的概率就是 $l_i/(k - j)$ 。所有旧页面的期望缺页次数就是:

$$\sum_{0 \leq j < k-l_i} \frac{l_i}{k-j} = l_i \sum_{l_i \leq j \leq k} \frac{1}{j} = l_i (H_k - H_j)$$

再加上新页面发生的缺页， $RM$ 总的开销不会超过：

$$Miss(RM, S) \leq 2H_k \times OPT(S) \leq 2 \log k \times OPT(S) + O(1)$$

**定理 3.2** 页面置换问题不存在 $c$ -Competitive的随机化算法，对于任何 $c < H_{n-1}$ 。

因此我们可以认为，在竞争比的意义下， $RM$ 几乎已经是最优的算法了。使用 $RM$ 可以保证在被某些程序“恶意攻击”时，页面换入换出的次数不会超过最优算法的 $2 \log k$ 倍；与之相比，任何确定性算法至多只能做到不超过 $k$ 倍。

## 4 Belady现象

### 参考文献

- [1] Belady, L. A . A study of replacement algorithms for a virtual-storage computer[J]. IBM Systems Journal, 1966, 5(2):78-101.
- [2] Fiat A , Karp R M , Luby M , et al. Competitive paging algorithms[J]. Journal of Algorithms, 1991, 12(4):685-699.
- [3] [https://en.wikipedia.org/wiki/Yao's\\_principle](https://en.wikipedia.org/wiki/Yao's_principle)
- [4] Sleator D D , Tarjan R E . Amortized efficiency of list update and paging rules[J]. Communications of the Acm, 1985, 28(2):202-208.
- [5] <https://piazza.com/class/i5j09fns17k5x0?cid=1296>
- [6] <https://riptutorial.com/algorithm/topic/8022/online-algorithms>
- [7] <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-854j-advanced-algorithms-fall-2005/lecture-notes/n23online.pdf>
- [8] [https://en.wikipedia.org/wiki/Page\\_replacement\\_algorithm#Conservative\\_algorithms](https://en.wikipedia.org/wiki/Page_replacement_algorithm#Conservative_algorithms)
- [9] <https://riptutorial.com/algorithm/example/25916/paging-online-caching->