# 华仔-技术博客

《面向对象葵花宝典》，写代码的架构师，做技术的管理者



个人资料


华仔爱技术

访问： 673205次

积分： 8873

等级： BLOG 6



## 缓存淘汰算法系列之2——LFU类

标签： 缓存 淘汰算法 LFU

2012-06-09 17:46 8119人阅读 评论(1) 收藏 举报

分类：

软件设计（78）



目录(?) [+]

# 1. LFU类

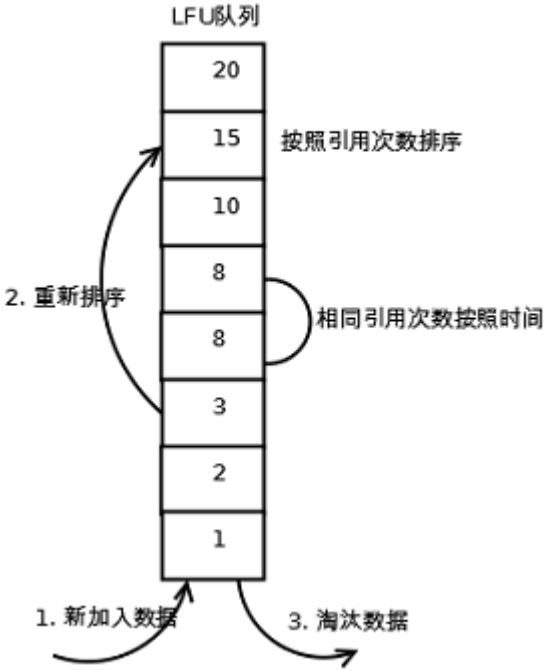
## 1.1. LFU

### 1.1.1. 原理

LFU（Least Frequently Used）算法根据数据的历史访问频率来淘汰数据，其核心思想是“如果数据过去被访问多次，那么将来被访问的频率也更高”。

### 1.1.2. 实现

LFU的每个数据块都有一个引用计数，所有数据块按照引用计数排序，具有相同引用计数的数据块则按照时间排序。
具体实现如下：




1. 新加入数据插入到队列尾部（因为引用计数为1）；
2. 队列中的数据被访问后，引用计数增加，队列重新排序；
3. 当需要淘汰数据时，将已经排序的列表最后的数据块删除。

### 1.1.3. 分析

### l 命中率

一般情况下，LFU效率要优于LRU，且能够避免周期性或者偶发性的操作导致缓存命中率下降的问题。但LFU需要记录数据的历史访问记录，一旦数据访问模式改变，LFU需要更长时间来适用新的访问模式，即：LFU存在历史数据影响将来数据的“缓存污染”效用。

**| 复杂度**

需要维护一个队列记录所有数据的访问记录，每个数据都需要维护引用计数。

**| 代价**

需要记录所有数据的访问记录，内存消耗较高；需要基于引用计数排序，性能消耗较高。

## 1.2. LFU*

### 1.2.1. 原理

基于LFU的改进算法，其核心思想是“只淘汰访问过一次的数据”。

### 1.2.2. 实现

LFU*数据缓存实现和LFU一样，不同的地方在于淘汰数据时，LFU*只淘汰引用计数为1的数据，且如果所有引用计数为1的数据大小之和都没有新加入的数据那么大，则不淘汰数据，新的数据也不缓存。

### 1.2.3. 分析

**| 命中率**

和LFU类似，但由于其不淘汰引用计数大于1的数据，则一旦访问模式改变，LFU*无法缓存新的数据，因此这个算法的应用场景比较有限。

**| 复杂度**

需要维护一个队列，记录引用计数为1的数据。

**| 代价**

相比LFU要低很多，不需要维护所有数据的历史访问记录，只需要维护引用次数为1的数据，也不需要排序。

## 1.3. LFU-Aging

### 1.3.1. 原理

基于LFU的改进算法，其核心思想是“除了访问次数外，还要考虑访问时间”。这样做的主要原因是解决LFU缓存污染的问题。

### 1.3.2. 实现

虽然LFU-Aging考虑时间因素，但其算法并不直接记录数据的访问时间，而是通过平均引用计数来标识时间。
LFU-Aging在LFU的基础上，增加了一个最大平均引用计数。当当前缓存中的数据“引用计数平均值”达到或者超过“最大平均引用计数”时，则将所有数据的引用计数都减少。减少的方法有多种，可以直接减为原来的一半，也可以减去固定的值等。

### 1.3.3. 分析

**l 命中率**
LFU-Aging的效率和LFU类似，当访问模式改变时，LFU-Aging能够更快的适用新的数据访问模式，效率要高。
**l 复杂度**
在LFU的基础上增加平均引用次数判断和处理。
**l 代价**
和LFU类似，当平均引用次数超过指定阈值（Aging）后，需要遍历访问列表。

## 1.4. LFU*-Aging

#### 1.4.1. 原理

LFU*和LFU-Aging的合成体。

#### 1.4.2. 实现

略。

#### 1.4.3. 分析

**l 命中率**
和LFU-Aging类似。
**l 复杂度**
比LFU-Aging简单一些，不需要基于引用计数排序。
**l 代价**
比LFU-Aging少一些，不需要基于引用计数排序。

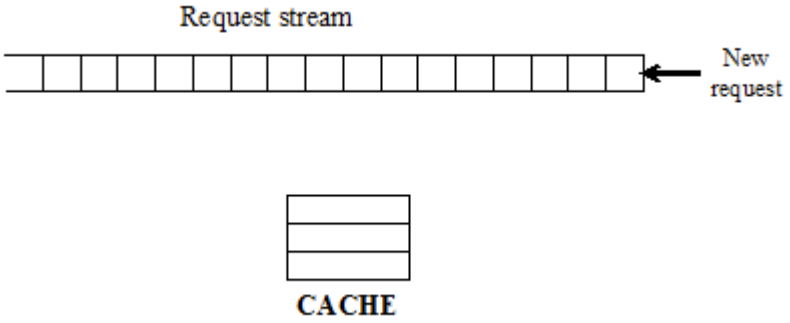
### 1.5. Window-LFU

#### 1.5.1. 原理

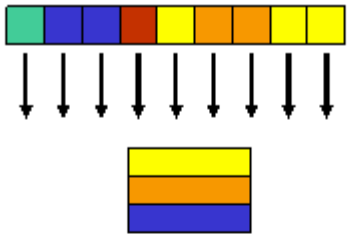
Windows-LFU是LFU的一个改进版，差别在于Window-LFU并不记录所有数据的访问历史，而只是记录过去一段时间内的访问历史，这就是Window的由来，基于这个原因，传统的LFU又被称为“Perfect-LFU”。

#### 1.5.2. 实现

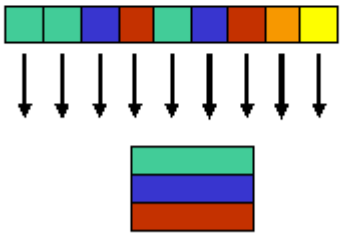
与LFU的实现基本相同，差别在于不需要记录所有数据的历史访问数据，而只记录过去一段时间内的访问历史。具体实现如下：




1）记录了过去W个访问记录；
2）需要淘汰时，将W个访问记录按照LFU规则排序淘汰
举例如下：
假设历史访问记录长度设为9，缓存大小为3，图中不同颜色代表针对不同数据块的访问，同一颜色代表针对同一数据的多次访问。



样例1



样例2

样例1：黄色访问3次，蓝色和橘色都是两次，橘色更新，因此缓存黄色、橘色、蓝色三个数据块

样例2：绿色访问3次，蓝色两次，暗红两次，蓝色更新，因此缓存绿色、蓝色、暗红三个数据块

### 1.5.3. 分析

**l 命中率**

Window-LFU的命中率和LFU类似，但Window-LFU会根据数据的访问模式而变化，能够更快的适应新的数据访问模式，”缓存污染“问题不严重。

**l 复杂度**

需要维护一个队列，记录数据的访问流历史；需要排序。

**l 代价**

Window-LFU只记录一部分的访问历史记录，不需要记录所有的数据访问历史，因此内存消耗和排序消耗都比LFU要低。

## 1.6. LFU类算法对比

由于不同的访问模型导致命中率变化较大，此处对比仅基于理论定性分析，不做定量分析。

| 对比点 | 对比 |
| --- | --- |
| 命中率 | Window-LFU/LFU-Aging > LFU*-Aging > LFU > LFU* |
| 复杂度 | LFU-Aging > LFU> LFU*-Aging >Window-LFU > LFU* |
| 代价 | LFU-Aging > LFU > Window-LFU > LFU*-Aging > LFU* |

上一篇 缓存淘汰算法系列之1——LRU类

下一篇 缓存淘汰算法系列之3——FIFO类

## 我的同类文章

### 软件设计（78）

- BAT解密：互联网技术发... 2016-08-01 阅读 1187
- 给飞驰的法拉利换引擎 - ... 2016-06-02 阅读 4710
- 给飞驰的法拉利换引擎 - ... 2016-05-24 阅读 4501
- BAT解密：互联网技术发... 2016-04-29 阅读 13768
- BAT解密：互联网技术发... 2015-12-31 阅读 1929
- BAT解密：互联网技术发... 2015-11-11 阅读 2902
- 异地多活设计辣么难？其... 2016-08-01 阅读 965
- 给飞驰的法拉利换引擎 - ... 2016-05-27 阅读 4782
- 给飞驰的法拉利换引擎 - ... 2016-05-18 阅读 2713
- 使用开源项目的正确姿势... 2016-03-03 阅读 3958
- 面向业务的立体化高可用... 2015-11-16 阅读 2078

## 猜你在找

Python算法实战视频课程--队列的应用
数据结构和算法
数据结构基础系列(1)：数据结构和算法
Python算法实战视频课程--栈的应用
数据结构基础系列(9)：排序
缓存淘汰算法系列之3FIFO类
缓存淘汰算法系列之1LRU类
缓存淘汰算法系列之3FIFO类
缓存淘汰算法系列之1LRU类
缓存淘汰算法系列之1LRU类

查看评论

1楼 在hust快乐的学习 2013-05-30 10:24发表

好文！

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker
OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC
WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML
LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra
CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App
SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP
HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap