CART for Linux

By

Rahul Iyer

Contents

- LRU and its problems
- How Linux does it
- Introducing CART
- Estimates and Issues
- And, Finally…

Problems with LRU

- Overhead of Moving to MRU position – use CLOCK
- Does not capture Frequency
- LFU can be used, but no Recency Information
- LFRU – The two taken together
- Uses Page aging

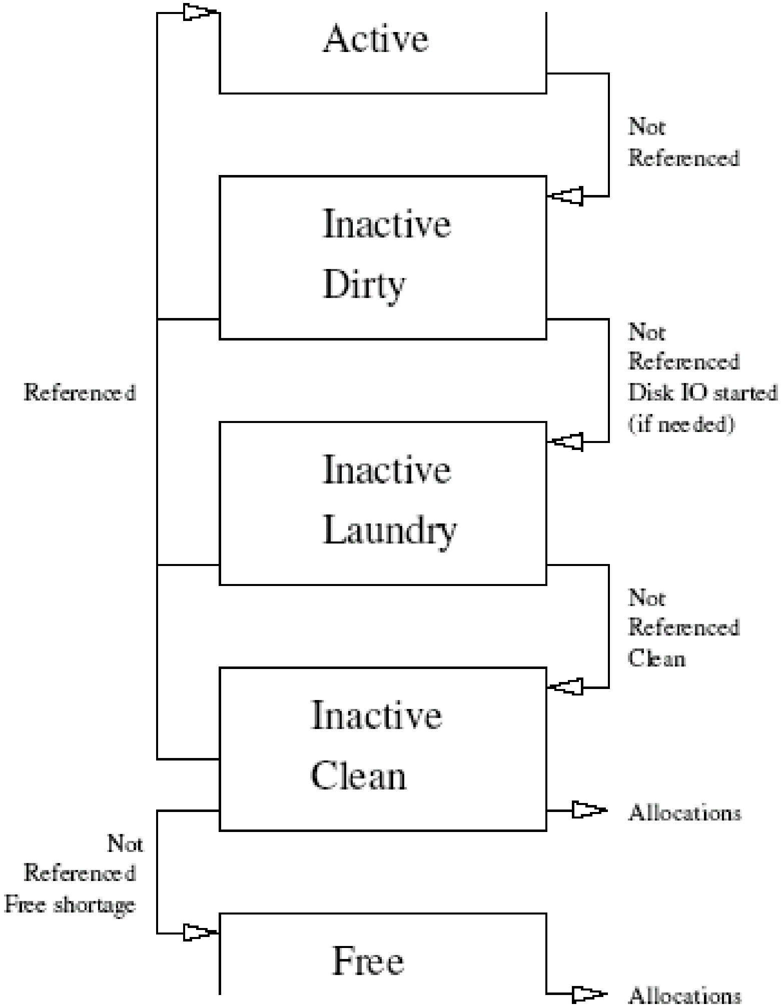
The Linux Way

- Uses the rmap VM by Rik van Riel
- Tries to capture Frequency using Page aging
- Borrowed from FreeBSD
- On a Hit, Age = Age + 2
- On a Miss, Age = Age/2
- Age = 0 implies ready for eviction

The Linux Way (Contd…)

- Unnecessary scanning
- Use Mach like Active and Inactive lists
- Split into more than one list – one for each stage of the page out process
- Active, Inactive Dirty, Inactive Laundry, Inactive Clean and Free

The Linux Way



Introducing CART

- Claims to be the best solution
- Uses 2 lists of size c
- Splits each list into 2 parts: the cache (T) and history (B)
- T1, B1 capture recency, T2, B2 capture frequency
- T1 + T2 <= c
- T1 + T2 + B1 + B2 <= 2c

Introducing CART (Contd…)

- T1 is ‘p’ elements long
- ‘p’ is initialized to 0, but adjusts as per access pattern
- On a cache miss, replace from T1 if T1>p else from T2
- If hit in B1, increase p, else decrease p
- Replacements within lists are LRU
- Can be approximated by CLOCK

Estimates and Issues

- CART => 300-400 lines of code
- Integration with existing VM is the major issue
- Possible modification of kswapd (100-200 lines)

Resources

- Riel R - Page Replacement in Linux 2.4 http://www.surriel.com/lectures/linux24-vm-freenix01.pdf
- Riel R - Towards an O(1) VM http://www.surriel.com/lectures/ols2003/riel2003_o1_vm.pdf
- D.S Modha et al. - One Up on LRU http://www.almaden.ibm.com/cs/people/dmodha/oneup.pdf
- D.S Modha et al. - CAR – Clock with Adaptive Replacement http://www.almaden.ibm.com/cs/people/dmodha/clockfast.pdf
- Gorman M. – Understanding the Linux Virtual Memory Manager http://www.csn.ul.ie/~mel/projects/vm/guide/pdf/understand.pdf
- Bovet D.P, Cesati M - Understanding the Linux Kernel