OS HW04: Memory Manager

資訊系 F74058037 鄭洋

2016年12月18日 星期日

1. Overview

本次作業要完成的page-fault service routine (memory manager)其流程可以歸納為:

(1) Find the location of the desired page on the disk.

(2) Find a free frame:

a. if there is a free frame, use it.

b. if there is no free frame, use a ***page-replacement algorithm*** to select a ***victim frame***.

c. write the victim frame to the disk; change the page and frame tables accordingly.

(3) Read the desired page into the newly freed frame; change the page and frame tables.

採取何種策略選擇victim frame是不同的page-replacement algorithms主要不同之處，其直接影響了最後的***page error rate***.

在此探討FIFO、LRU、Random等幾種page-replacement algorithm之區別與優缺點.

2. First-in-First-out (FIFO)

2.1 description

當memory已滿，有page必須被替換時，FIFO會選擇將the oldest page，也就是the first one paged in, 拿出memory並放回disk.

可以採取***FIFO queue***來記錄在memory中的pages. 選victim時，queue head就會被選中.

2.2 pros and cons

pros:

(1) 模型簡單, low-overhead implementation

cons:

(1) 有可能做壞的選擇，比如first-in page在一段時間後再次被叫到，而它已經被踢到disk了

(2) RAM越大不代表fault越少 (Bélády's anomaly)

e.g. #virtual page=5, Access 123412512345
#physical frame=3, #page fault=9;
#physical frame=4, #page fault=10;

3. Least-Recently-Use (LRU)

3.1 description

LRU(以及LFU、MRU)的思想是，儘管我們不能預測未來發生的事情，但是可以用過去的情況來估計未來.

LRU的假設是，近期用到的page，在未來也可以能會被用到. 因此，在選取victim時，LRU會選取過去最久沒用到的那個page. LRU可以用doubly-linked list來實現.

3.2 pros and cons

pros:

(1) 是對OPT很好的一個模擬

(2) 不會有Bélády's anomaly的問題

(3) 對於持有temporal locality特性的program, LRU應該會有很好的表現, 見table 3.2.1紅色高亮處

	FIFO	LRU	Random
e.r. best	0.312	**0.273**	0.338
e.r. worse	0.491	0.477	0.419

table 3.2.1

table 3.2.1 注： *e.r. = error rate,* 數據來源*"/hw4_trace/temporal_locality"*， *random*採取的是*rand(),(pseudo random)*

cons:

(1) 比FIFO實現起來要困難許多

(2) 由於對linked list要時常(delete，insert), 很慢，時間消耗大, 需要將這個放在cache裡做.

(3) 在非常極端的情況，表現會較差，見table 4.2.1紅色高亮處

	FIFO	LRU	Random
e.r. best	0.267	0.213	0.267
e.r. worse	0.556	**0.981**	0.389

table 4.2.1

table 4.2.1 注：*e.r. = error rate,* 數據來源*"/hw4_trace/spacial_locality"*，*random*採取的是*rand(),(pseudo random)*

4. Random

4.1 description

隨機選一個victim

4.2 pros and cons

pros:

(1) 模型簡單，low-overhead implementation

(2) 表現比較穩定, 因為每次選victim都是隨機選的，所以在不同情況下，表現差異都不會有太大差距，也不會太糟糕. 見table 3.2.1和table 4.2.1綠色框框

cons:

(1) 無法逼近OPT，沒有針對性的使用場景

5. Summary

將三者作比較:

	FIFO	LRU	Random
victim	最早進入的	最久沒用的	隨機
Bélády's anomaly	是	否	是
implement	容易	複雜	容易

可以看到，三者之間其實只有LRU是考慮了過去的使用情況的，FIFO雖然有記錄page in的情況，但是不會在最近被使用過後更新queue.
並且只有LRU滿足：RAM大，則fault少.

我猜測這兩者是LRU最為常用的主要原因.

總的來說，LRU看起來比FIFO表現要好，但是也存在一些情況，LRU的Miss率會遠高於其他策略.

6. Reference

1. "Operating System Concepts, 10th edition"
2. http://www.cs.cornell.edu/courses/cs4410/2015su/lectures/lec15-replacement.html
3. https://en.wikipedia.org/wiki/B%C3%A9l%C3%A1dy's_anomaly