Georgios Karagiannis
Roman Sodermans
Michael Tang
Calvin Yang
3/1/2017

CMPS111-02: Design Document Assignment 3

Altered files:

sys/vm:

- vm_pageout.c
- vm_page.c
- vm_phys.c

Changes made in files

Changes made in sys/vm/vm_pageout.c:

We created counters (done in vm_pageout_scan) to keep track of:

1. Number of pages in Active queue
2. Number of pages in Inactive queue
3. Which queues were scanned in the run (Active or Inactive)
4. Number of pages moved from Active to Inactive list
5. Number of pages moved from Inactive to Cache/Free list
6. Number of pages queued for flush

With these counters we found where in the vm_pageout_scan function these events are occurring and incremented our counters accordingly. For the number of pages in active and inactive queues, we just referenced a pre-existing variable that keeps track of these numbers. These counters are used to compare the old-daemon algorithm statistics with our slim chance algorithm.

The Pageout scan was made to run more frequently. It now runs every 10 seconds instead of 10 minutes. We did this by:

- Changing variable vm_pageout_update_period = 10 seconds (was originally 600)

This is done to implement the slim chance algorithm.

At every scan, activity count is now halved instead of decremented. This is done to implement the slim chance algorithm. Because we are halving the activity count instead of decrementing, the activity count will go below its. We did this by:

- Changing p->act_count = p->act_count /2 in function vm_pageout_scan;

Changes made in sys/vm/vm_page.c:

In vm_page.c, we moved incoming pages to the head of the active list instead of the tail. To do this, we modified the line:

- TAILQ_INSERT_TAIL to TAILQ_INSERT_HEAD

This modification was done in three functions, vm_page_enqueue(), vm_page_requeue(), vm_page_requeue(), and vm_page_requeue_locked(). We do this to implement the slim chance algorithm. Inserting pages at the head instead of the tail means they must be considered immediately to move.

Changes made in sys/vm/vm_phys.c:

We add to head instead of tail, as it was initially for the free pages, so the page moved to free list is considered immediately. We did this by:

- Changing vm_phys_free_pages(). Instead of adding to tail, we add to head.

We changed the arguments of the vm_freelist_add() function call from its original 1 to 0. We didn't change the function definition, as the function is called from elsewhere in the code for other purposes; we only changed this specific function call, which is done when adding a page to the free list.