

[PATCH 8/9] clockpro-rename_PG_active.patch

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-12/msg08499.html>

- *From:* Peter Zijlstra <a.p.zijlstra@xxxxxxxxxx>
 - *Date:* Fri, 30 Dec 2005 23:43:54 +0100
-

From: Peter Zijlstra <a.p.zijlstra@xxxxxxxxxx>

New semantics, new name. Since the semantics of PG_activate changed drastically with the clockpro code, change its name.

Signed-off-by: Peter Zijlstra <a.p.zijlstra@xxxxxxxxxx>

```
fs/exec.c | 2 +-
include/linux/mm_page_replace.h | 4 +++-
include/linux/page-flags.h | 12 +++++-----
mm/clockpro.c | 22 ++++++++-----
mm/hugetlb.c | 2 +-
mm/memory.c | 6 +++---
mm/page_alloc.c | 6 +++---
mm/swap.c | 2 +-
mm/swap_state.c | 2 +-
mm/vmscan.c | 2 +-
10 files changed, 30 insertions(+), 30 deletions(-)
```

Index: linux-2.6-git/fs/exec.c

```
-----
--- linux-2.6-git.orig/fs/exec.c
+++ linux-2.6-git/fs/exec.c
@@ -321,7 +321,7 @@ void install_arg_page(struct vm_area_str
goto out;
}
inc_mm_counter(mm, anon_rss);
- SetPageActive(page);
+ SetPageHot(page);
lru_cache_add(page);
set_pte_at(mm, address, pte, pte_mkdirty(pte_mkwrite(mk_pte(
page, vma->vm_page_prot))));
Index: linux-2.6-git/include/linux/mm_page_replace.h
```

```
-----
--- linux-2.6-git.orig/include/linux/mm_page_replace.h
+++ linux-2.6-git/include/linux/mm_page_replace.h
@@ -61,7 +61,7 @@ void page_replace_remember(struct zone *
static inline
```

[PATCH 8/9] clockpro-rename_PG_active.patch

```
void __page_replace_rotate_reclaimable(struct zone *zone, struct page *page)
{
- if (PageLRU(page) && !PageActive(page)) {
+ if (PageLRU(page) && !PageHot(page)) {
list_move_tail(&page->lru, &zone->list_hand[hand_cold]);
inc_page_state(pgrotated);
}
@@ -72,7 +72,7 @@ del_page_from_lru(struct zone *zone, str
{
list_del(&page->lru);
--zone->nr_resident;
- if (!TestClearPageActive(page))
+ if (!TestClearPageHot(page))
--zone->nr_cold;
}
```

Index: linux-2.6-git/include/linux/page-flags.h

```
----- linux-2.6-git.orig/include/linux/page-flags.h
+++ linux-2.6-git/include/linux/page-flags.h
@@ -58,7 +58,7 @@
```

```
#define PG_dirty 4
#define PG_lru 5
-#define PG_active 6
+#define PG_hot 6
#define PG_slab 7 /* slab debug (Suparna wants this) */

#define PG_checked 8 /* kill me in 2.5.<early>. */
@@ -205,11 +205,11 @@ extern void __mod_page_state(unsigned lo
#define TestSetPageLRU(page) test_and_set_bit(PG_lru, &(page)->flags)
#define TestClearPageLRU(page) test_and_clear_bit(PG_lru, &(page)->flags)

-#define PageActive(page) test_bit(PG_active, &(page)->flags)
-#define SetPageActive(page) set_bit(PG_active, &(page)->flags)
-#define ClearPageActive(page) clear_bit(PG_active, &(page)->flags)
-#define TestClearPageActive(page) test_and_clear_bit(PG_active, &(page)->flags)
-#define TestSetPageActive(page) test_and_set_bit(PG_active, &(page)->flags)
+#define PageHot(page) test_bit(PG_hot, &(page)->flags)
+#define SetPageHot(page) set_bit(PG_hot, &(page)->flags)
+#define ClearPageHot(page) clear_bit(PG_hot, &(page)->flags)
+#define TestClearPageHot(page) test_and_clear_bit(PG_hot, &(page)->flags)
+#define TestSetPageHot(page) test_and_set_bit(PG_hot, &(page)->flags)

#define PageSlab(page) test_bit(PG_slab, &(page)->flags)
#define SetPageSlab(page) set_bit(PG_slab, &(page)->flags)
Index: linux-2.6-git/mm/clockpro.c
```

```
----- linux-2.6-git.orig/mm/clockpro.c
+++ linux-2.6-git/mm/clockpro.c
@@ -127,7 +127,7 @@ void __select_list_hand(struct zone *zon
```

[PATCH 8/9] clockpro-rename_PG_active.patch

[PATCH 8/9] clockpro-rename_PG_active.patch

```
* Insert page into @zones clock and update adaptive parameters.
*
* Several page flags are used for insertion hints:
- * PG_active – insert as an active page
+ * PG_hot – insert as an active page
* PG_test – use the use-once logic
*
* For now we will ignore the active hint; the use once logic is
@@ -142,8 +142,8 @@ void __page_replace_insert(struct zone *
rflags = nonresident_get(page_mapping(page), page_index(page));

- /* ignore the PG_active hint */
- ClearPageActive(page);
+ /* ignore the PG_hot hint */
+ ClearPageHot(page);

/* abuse the PG_test flag for pagecache use-once */
if (!TestClearPageTest(page)) {
@@ -153,7 +153,7 @@ void __page_replace_insert(struct zone *
* ie. right behind Hcold.
*/
if (rflags & NR_found) {
- SetPageActive(page);
+ SetPageHot(page);
__cold_target_inc(zone, 1);
} else {
SetPageTest(page);
@@ -233,7 +233,7 @@ static int isolate_lru_pages(struct zone
} else {
list_add(&page->lru, dst);
nr_taken++;
- if (!PageActive(page))
+ if (!PageHot(page))
--zone->nr_cold;
}
}
@@ -258,7 +258,7 @@ static void __page_release(struct zone *
{
if (TestSetPageLRU(page))
BUG();
- if (!PageActive(page))
+ if (!PageHot(page))
++zone->nr_cold;
++zone->nr_resident;

@@ -311,14 +311,14 @@ void page_replace_activate(struct page *
{
int hot, test;

- hot = PageActive(page);
```

[PATCH 8/9] clockpro-rename_PG_active.patch

```
+ hot = PageHot(page);
test = PageTest(page);

if (hot) {
BUG_ON(test);
} else {
if (test) {
- SetPageActive(page);
+ SetPageHot(page);
/*
* Leave PG_test set for new hot pages in order to
* recognise then in reinsert() and do accounting.
@@ -358,7 +358,7 @@ void page_replace_reinsert(struct zone *
struct page *page = lru_to_page(page_list);
prefetchw_prev_lru_page(page, page_list, flags);

- if (PageActive(page) && PageTest(page)) {
+ if (PageHot(page) && PageTest(page)) {
ClearPageTest(page);
++dct;
}
@@ -515,7 +515,7 @@ static void rotate_hot(struct zone *zone
struct page *page = lru_to_page(&l_hold);
prefetchw_prev_lru_page(page, &l_hold, flags);

- if (PageActive(page)) {
+ if (PageHot(page)) {
BUG_ON(PageTest(page));

/*
@@ -528,7 +528,7 @@ static void rotate_hot(struct zone *zone
if (/*(((reclaim_mapped && mapped) || !mapped) ||
(total_swap_pages == 0 && PageAnon(page))) && */
!page_referenced(page, 0, 1)) {
- ClearPageActive(page);
+ ClearPageHot(page);
++pgdeactivate;
}
}
```

Index: linux-2.6-git/mm/hugetlb.c

```
-----
--- linux-2.6-git.orig/mm/hugetlb.c
+++ linux-2.6-git/mm/hugetlb.c
@@ -145,7 +145,7 @@ static void update_and_free_page(struct
nr_huge_pages_node[page_zone(page)->zone_pgdat->node_id]--;
for (i = 0; i < (HPAGE_SIZE / PAGE_SIZE); i++) {
page[i].flags &= ~(1 << PG_locked | 1 << PG_error | 1 << PG_referenced |
- 1 << PG_dirty | 1 << PG_active | 1 << PG_reserved |
+ 1 << PG_dirty | 1 << PG_hot | 1 << PG_reserved |
1 << PG_private | 1 << PG_writeback);
set_page_count(&page[i], 0);
```

[PATCH 8/9] clockpro-rename_PG_active.patch

}

Index: linux-2.6-git/mm/memory.c

```

=====
--- linux-2.6-git.orig/mm/memory.c
+++ linux-2.6-git/mm/memory.c
@@ -1521,7 +1521,7 @@ gotten:
ptep_establish(vma, address, page_table, entry);
update_mmu_cache(vma, address, entry);
lazy_mmu_prot_update(entry);
- SetPageActive(new_page);
+ SetPageHot(new_page);
lru_cache_add(new_page);
page_add_anon_rmap(new_page, vma, address);

```

```

@@ -1978,7 +1978,7 @@ static int do_anonymous_page(struct mm_s
if (!pte_none(*page_table))
goto release;
inc_mm_counter(mm, anon_rss);
- SetPageActive(page);
+ SetPageHot(page);
lru_cache_add(page);
SetPageReferenced(page);
page_add_anon_rmap(page, vma, address);
@@ -2111,7 +2111,7 @@ retry:
set_pte_at(mm, address, page_table, entry);
if (anon) {
inc_mm_counter(mm, anon_rss);
- SetPageActive(new_page);
+ SetPageHot(new_page);
lru_cache_add(new_page);
page_add_anon_rmap(new_page, vma, address);
} else {
Index: linux-2.6-git/mm/page_alloc.c

```

```

=====
--- linux-2.6-git.orig/mm/page_alloc.c
+++ linux-2.6-git/mm/page_alloc.c
@@ -136,7 +136,7 @@ static void bad_page(const char *functio
page->flags &= ~(1 << PG_lru |
1 << PG_private |
1 << PG_locked |
- 1 << PG_active |
+ 1 << PG_hot |
1 << PG_dirty |
1 << PG_reclaim |
1 << PG_slab |
@@ -344,7 +344,7 @@ static inline int free_pages_check(const
1 << PG_lru |
1 << PG_private |
1 << PG_locked |
- 1 << PG_active |
+ 1 << PG_hot |

```

[PATCH 8/9] clockpro-rename_PG_active.patch

```
1 << PG_reclaim |
1 << PG_slab |
1 << PG_swapcache |
@@ -481,7 +481,7 @@ static int prep_new_page(struct page *pa
1 << PG_lru |
1 << PG_private |
1 << PG_locked |
- 1 << PG_active |
+ 1 << PG_hot |
1 << PG_dirty |
1 << PG_reclaim |
1 << PG_slab |
Index: linux-2.6-git/mm/swap.c
```

```
----- linux-2.6-git.orig/mm/swap.c
+++ linux-2.6-git/mm/swap.c
@@ -75,7 +75,7 @@ int rotate_reclaimable_page(struct page
return 1;
if (PageDirty(page))
return 1;
- if (PageActive(page))
+ if (PageHot(page))
return 1;
if (!PageLRU(page))
return 1;
Index: linux-2.6-git/mm/swap_state.c
```

```
----- linux-2.6-git.orig/mm/swap_state.c
+++ linux-2.6-git/mm/swap_state.c
@@ -353,7 +353,7 @@ struct page *read_swap_cache_async(swp_e
/*
* Initiate read into locked page and return.
*/
- SetPageActive(new_page);
+ SetPageHot(new_page);
lru_cache_add(new_page);
swap_readpage(NULL, new_page);
return new_page;
Index: linux-2.6-git/mm/vmscan.c
```

```
----- linux-2.6-git.orig/mm/vmscan.c
+++ linux-2.6-git/mm/vmscan.c
@@ -339,7 +339,7 @@ static try_pageout_t try_pageout(struct
int may_enter_fs;
int referenced;

- if (PageActive(page))
+ if (PageHot(page))
goto keep;

if (TestSetPageLocked(page))
```

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>

- **References:**

- ◆ **[PATCH] vm: page-replace and clockpro**

- ◆ *From:* Peter Zijlstra

- Prev by Date: **[PATCH 03/14] page-replace-remove-sc-from-refill.patch**

- Next by Date: **[PATCH 7/9] clockpro-remove-old.patch**

- Previous by thread: **[PATCH 03/14] page-replace-remove-sc-from-refill.patch**

- Next by thread: **[PATCH 7/9] clockpro-remove-old.patch**

- Index(es):

- ◆ **Date**

- ◆ **Thread**