

[PATCH 05/14] page-replace-remove-loop.patch

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-12/msg08482.html>

- *From:* Peter Zijlstra <a.p.zijlstra@xxxxxxxx>
 - *Date:* Fri, 30 Dec 2005 23:41:04 +0100
-

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

This patch removes the loop in shrink_cache() by directly taking sc->nr_to_scan pages.

Kudos to Wu Fengguang who did a similar patch.

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

mm/vmscan.c | 88 ++++++-----
1 files changed, 41 insertions(+), 47 deletions(-)

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

```
=====
--- linux-2.6-git.orig/mm/vmscan.c 2005-12-10 17:13:57.000000000 +0100
+++ linux-2.6-git/mm/vmscan.c 2005-12-10 18:19:30.000000000 +0100
@@ -653,61 +653,55 @@ static void shrink_cache(struct zone *zo
{
LIST_HEAD(page_list);
struct pagevec pvec;
- int max_scan = sc->nr_to_scan;
-
- pagevec_init(&pvec, 1);
+ struct page *page;
+ int nr_taken;
+ int nr_scan;
+ int nr_freed;

lru_add_drain();
+
spin_lock_irq(&zone->lru_lock);
- while (max_scan > 0) {
- struct page *page;
- int nr_taken;
- int nr_scan;
- int nr_freed;
-
- nr_taken = isolate_lru_pages(sc->swap_cluster_max,
```

[PATCH 05/14] page-replace-remove-loop.patch

```
- &zone->inactive_list,
- &page_list, &nr_scan);
- zone->nr_inactive -= nr_taken;
- zone->pages_scanned += nr_scan;
- spin_unlock_irq(&zone->lru_lock);
+ nr_taken = isolate_lru_pages(sc->nr_to_scan,
+ &zone->inactive_list,
+ &page_list, &nr_scan);
+ zone->nr_inactive -= nr_taken;
+ zone->pages_scanned += nr_scan;
+ spin_unlock_irq(&zone->lru_lock);

- if (nr_taken == 0)
- goto done;
+ if (nr_taken == 0)
+ return;

- max_scan -= nr_scan;
- if (current_is_kswapd())
- mod_page_state_zone(zone, pgscan_kswapd, nr_scan);
- else
- mod_page_state_zone(zone, pgscan_direct, nr_scan);
- nr_freed = shrink_list(&page_list, sc);
- if (current_is_kswapd())
- mod_page_state(kswapd_steal, nr_freed);
- mod_page_state_zone(zone, pgsteal, nr_freed);
- sc->nr_to_reclaim -= nr_freed;
+ if (current_is_kswapd())
+ mod_page_state_zone(zone, pgscan_kswapd, nr_scan);
+ else
+ mod_page_state_zone(zone, pgscan_direct, nr_scan);
+ nr_freed = shrink_list(&page_list, sc);
+ if (current_is_kswapd())
+ mod_page_state(kswapd_steal, nr_freed);
+ mod_page_state_zone(zone, pgsteal, nr_freed);
+ sc->nr_to_reclaim -= nr_freed;

- spin_lock_irq(&zone->lru_lock);
- /*
- * Put back any unfreeable pages.
- */
- while (!list_empty(&page_list)) {
- page = lru_to_page(&page_list);
- if (TestSetPageLRU(page))
- BUG();
- list_del(&page->lru);
- if (PageActive(page))
- add_page_to_active_list(zone, page);
- else
- add_page_to_inactive_list(zone, page);
- if (!pagevec_add(&pvec, page)) {
```

[PATCH 05/14] page-replace-remove-loop.patch

```
- spin_unlock_irq(&zone->lru_lock);
- __pagevec_release(&pvec);
- spin_lock_irq(&zone->lru_lock);
- }
+ /*
+ * Put back any unfreeable pages.
+ */
+ pagevec_init(&pvec, 1);
+ spin_lock_irq(&zone->lru_lock);
+ while (!list_empty(&page_list)) {
+ page = lru_to_page(&page_list);
+ if (TestSetPageLRU(page))
+ BUG();
+ list_del(&page->lru);
+ if (PageActive(page))
+ add_page_to_active_list(zone, page);
+ else
+ add_page_to_inactive_list(zone, page);
+ if (!pagevec_add(&pvec, page)) {
+ spin_unlock_irq(&zone->lru_lock);
+ __pagevec_release(&pvec);
+ spin_lock_irq(&zone->lru_lock);
+ }
- }
+ }
spin_unlock_irq(&zone->lru_lock);
-done:
pagevec_release(&pvec);
}
```

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 10/14] page-replace-remove-mm inline.patch**
- Next by Date: **[PATCH 6/9] clockpro-clockpro.patch**
- Previous by thread: **[PATCH 10/14] page-replace-remove-mm inline.patch**
- Next by thread: **[PATCH 6/9] clockpro-clockpro.patch**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**