Subject: [PATCH v4 16/25] don't do __ClearPageSlab before freeing slab page.
Posted by Glauber Costa on Mon, 18 Jun 2012 10:28:09 GMT
View Forum Message <> Reply to Message

This will give the oportunity to the page allocator to
determine that a given page was previously a slab page, and
take action accordingly.

If memcg kmem is present, this means that that page needs to
be unaccounted. The page allocator will now have the responsibility
to clear that bit upon free_pages().

It is not uncommon to have the page allocator to check page flags.
Mlock flag, for instance, is checked pervasively all over the place.
So I hope this is okay for the slab as well.

Signed-off-by: Glauber Costa <glommer@parallels.com>
CC: Pekka Enberg <penberg@cs.helsinki.fi>
CC: Michal Hocko <mhocko@suse.cz>
CC: Kamezawa Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
CC: Johannes Weiner <hannes@cmpxchg.org>
CC: Suleiman Souhlal <suleiman@google.com>
---
 mm/page_alloc.c | 5 ++++-
 mm/slab.c | 5 -----
 mm/slob.c | 1 -
 mm/slub.c | 1 -
 4 files changed, 4 insertions(+), 8 deletions(-)

diff --git a/mm/page_alloc.c b/mm/page_alloc.c
index 918330f..a884a9c 100644
--- a/mm/page_alloc.c
+++ b/mm/page_alloc.c
@@ -697,8 +697,10 @@ static bool free_pages_prepare(struct page *page, unsigned int order)

  if (PageAnon(page))
   page->mapping = NULL;
- for (i = 0; i < (1 << order); i++)
+ for (i = 0; i < (1 << order); i++) {
+ __ClearPageSlab(page + i);
   bad += free_pages_check(page + i);
+ }
  if (bad)
   return false;

@@ -2505,6 +2507,7 @@ EXPORT_SYMBOL(get_zeroed_page);
 void __free_pages(struct page *page, unsigned int order)
 {

```
  if (put_page_testzero(page)) {
+  __ClearPageSlab(page);
   if (order == 0)
    free_hot_cold_page(page, 0);
   else
diff --git a/mm/slab.c b/mm/slab.c
index c548666..e537406 100644
--- a/mm/slab.c
+++ b/mm/slab.c
@@ -1795,11 +1795,6 @@ static void kmem_freepages(struct kmem_cache *cachep, void
*addr)
  else
   sub_zone_page_state(page_zone(page),
     NR_SLAB_UNRECLAIMABLE, nr_freed);
- while (i--) {
-  BUG_ON(!PageSlab(page));
-  __ClearPageSlab(page);
-  page++;
- }
  if (current->reclaim_state)
   current->reclaim_state->reclaimed_slab += nr_freed;
  free_pages((unsigned long)addr, cachep->gfporder);
diff --git a/mm/slob.c b/mm/slob.c
index 61b1845..b03d65e 100644
--- a/mm/slob.c
+++ b/mm/slob.c
@@ -360,7 +360,6 @@ static void slob_free(void *block, int size)
   if (slob_page_free(sp))
    clear_slob_page_free(sp);
   spin_unlock_irqrestore(&slob_lock, flags);
-  __ClearPageSlab(sp);
   reset_page_mapcount(sp);
   slob_free_pages(b, 0);
   return;
diff --git a/mm/slub.c b/mm/slub.c
index e685cfa..69c5677 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -1399,7 +1399,6 @@ static void __free_slab(struct kmem_cache *s, struct page *page)
   NR_SLAB_RECLAIMABLE : NR_SLAB_UNRECLAIMABLE,
   -pages);

- __ClearPageSlab(page);
  reset_page_mapcount(page);
  if (current->reclaim_state)
   current->reclaim_state->reclaimed_slab += pages;
--
1.7.10.2
```