
Subject: [PATCH v3 02/16] slub: use free_page instead of put_page for freeing kmalloc allocation

Posted by [Glauber Costa](#) on Tue, 18 Sep 2012 14:11:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

When freeing objects, the slub allocator will most of the time free empty pages by calling `__free_pages()`. But high-order kmalloc will be disposed by means of `put_page()` instead. It makes no sense to call `put_page()` in kernel pages that are provided by the object allocators, so we shouldn't be doing this ourselves. Aside from the consistency change, we don't change the flow too much. `put_page()`'s would call its dtor function, which is `__free_pages`. We also already do all of the Compound page tests ourselves, and the Mlock test we lose don't really matter.

[v2: modified Changelog]

Signed-off-by: Glauber Costa <glommer@parallels.com>

Acked-by: Christoph Lameter <cl@linux.com>

CC: David Rientjes <rientjes@google.com>

CC: Pekka Enberg <penberg@kernel.org>

mm/slub.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

diff --git a/mm/slub.c b/mm/slub.c

index 9f86353..09a91d0 100644

--- a/mm/slub.c

+++ b/mm/slub.c

```
@@ -3451,7 +3451,7 @@ void kfree(const void *x)
     if (unlikely(!PageSlab(page))) {
         BUG_ON(!PageCompound(page));
         kmemleak_free(x);
-    put_page(page);
+    __free_pages(page, compound_order(page));
     return;
     }
     slab_free(page->slab, page, object, _RET_IP_);
```

--

1.7.11.4
