Subject: [PATCH 1/3] slab: Proper off-slabs handling when duplicating caches
Posted by Anton Vorontsov on Mon, 30 Apr 2012 10:01:21 GMT
View Forum Message <> Reply to Message

OFF_SLAB is not CREATE_MASK bit, so we should clear it before
calling __kmem_cache_create(), otherwise kernel gets very upset,
see below.

As a side effect, now we let slab to reevaluate off-slab
decision, but the decision will be the same, because whether
we do off-slabs only depend on the size and create_mask
bits.

```
------------[ cut here ]------------
kernel BUG at mm/slab.c:2376!
invalid opcode: 0000 [#1] SMP
CPU 0
Pid: 14, comm: kworker/0:1 Not tainted 3.4.0-rc4+ #32 Bochs Bochs
RIP: 0010:[<ffffffff810c1839>]  [<ffffffff810c1839>] __kmem_cache_create+0x609/0x650
RSP: 0018:ffff8800072c9c90  EFLAGS: 00010286
RAX: 0000000000000800 RBX: ffffffff81f26bf8 RCX: 000000000000000b
RDX: 000000000000000c RSI: 000000000000000b RDI: ffff8800065c66f8
RBP: ffff8800072c9d40 R08: ffffffff80002800 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000000001 R12: ffff8800072c8000
R13: ffff8800072c9fd8 R14: ffffffffffffffff R15: ffff8800072c9d0c
FS:  00007f45eb0f2700(0000) GS:ffff880007c00000(0000) knlGS:0000000000000000
CS:  0010 DS: 0000 ES: 0000 CR0: 000000008005003b
CR2: ffffffffff600400 CR3: 000000000650e000 CR4: 00000000000006b0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400
Process kworker/0:1 (pid: 14, threadinfo ffff8800072c8000, task ffff88000725d100)
Stack:
 ffff8800072c9cb0 0000000000000000 ffffc9000000c000 ffffffff81621e80
 ffff8800072c9cc0 ffffffff81621e80 ffff8800072c9d40 ffffffff81355cbf
 ffffffff810c1944 0000000000000000 ffffffff81621ec0 ffffffff80002800
Call Trace:
 [<ffffffff81355cbf>] ? mutex_lock_nested+0x26f/0x340
 [<ffffffff810c1944>] ? kmem_cache_dup+0x44/0x110
 [<ffffffff810c2aa0>] ? memcg_create_kmem_cache+0xd0/0xd0
 [<ffffffff810c196b>] kmem_cache_dup+0x6b/0x110
 [<ffffffff810c2a73>] memcg_create_kmem_cache+0xa3/0xd0
 [<ffffffff810c2b1a>] memcg_create_cache_work_func+0x7a/0xe0
 [<ffffffff810405d4>] process_one_work+0x174/0x450
 [<ffffffff81040576>] ? process_one_work+0x116/0x450
 [<ffffffff81040e53>] worker_thread+0x123/0x2d0
 [<ffffffff81040d30>] ? manage_workers.isra.27+0x120/0x120
 [<ffffffff8104639e>] kthread+0x8e/0xa0
```

```
Signed-off-by: Anton Vorontsov <anton.vorontsov@linaro.org>
---
 mm/slab.c |    7 +++++++
 1 file changed, 7 insertions(+)

diff --git a/mm/slab.c b/mm/slab.c
index eed72ac..dff87ef 100644
--- a/mm/slab.c
+++ b/mm/slab.c
@@ -2619,6 +2619,13 @@ kmem_cache_dup(struct mem_cgroup *memcg, struct kmem_cache
*cachep)
   return NULL;

  flags = cachep->flags & ~SLAB_PANIC;
+ /*
+  * OFF_SLAB is not CREATE_MASK bit, so we should clear it before
+  * calling __kmem_cache_create(). As a side effect, we let slab
+  * to reevaluate off-slab decision; but that is OK, as the bit
+  * is automatically set depending on the size and other flags.
+  */
+ flags &= ~CFLGS_OFF_SLAB;
  mutex_lock(&cache_chain_mutex);
  new = __kmem_cache_create(memcg, name, obj_size(cachep),
      cachep->memcg_params.orig_align, flags, cachep->ctor);
--
1.7.9.2
```