Subject: [PATCH 14/23] slub: provide kmalloc_no_account
Posted by Glauber Costa on Sun, 22 Apr 2012 23:53:31 GMT
View Forum Message <> Reply to Message

Some allocations need to be accounted to the root memcg regardless
of their context. One trivial example, is the allocations we do
during the memcg slab cache creation themselves. Strictly speaking,
they could go to the parent, but it is way easier to bill them to
the root cgroup.

Only generic kmalloc allocations are allowed to be bypassed.

The function is not exported, because drivers code should always
be accounted.

Signed-off-by: Glauber Costa <glommer@parallels.com>
CC: Christoph Lameter <cl@linux.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>
---
 include/linux/slub_def.h | 1 +
 mm/slub.c | 21 +++++++++++++++++++++
 2 files changed, 22 insertions(+), 0 deletions(-)

diff --git a/include/linux/slub_def.h b/include/linux/slub_def.h
index 5f5e942..9a8000a 100644
--- a/include/linux/slub_def.h
+++ b/include/linux/slub_def.h
@@ -221,6 +221,7 @@ static __always_inline struct kmem_cache *kmalloc_slab(size_t size)
 }

 void *kmem_cache_alloc(struct kmem_cache *, gfp_t);
+void *kmalloc_no_account(size_t size, gfp_t);
 void *__kmalloc(size_t size, gfp_t flags);

 static __always_inline void *
diff --git a/mm/slub.c b/mm/slub.c
index 2285a96..d754b06 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -3359,6 +3359,27 @@ void *__kmalloc(size_t size, gfp_t flags)
 }
 EXPORT_SYMBOL(__kmalloc);

+void *kmalloc_no_account(size_t size, gfp_t flags)

```
+{
+ struct kmem_cache *s;
+ void *ret;
+
+ if (unlikely(size > SLUB_MAX_SIZE))
+  return kmalloc_large(size, flags);
+
+ s = get_slab(size, flags);
+
+ if (unlikely(ZERO_OR_NULL_PTR(s)))
+  return s;
+
+ ret = slab_alloc(s, flags, NUMA_NO_NODE, _RET_IP_);
+
+ trace_kmalloc(_RET_IP_, ret, size, s->size, flags);
+
+ return ret;
+}
+EXPORT_SYMBOL(kmalloc_no_account);
+
 #ifdef CONFIG_NUMA
 static void *kmalloc_large_node(size_t size, gfp_t flags, int node)
 {
--
1.7.7.6
```