Subject: Re: [PATCH v5 11/18] sl[au]b: Allocate objects from memcg cache
Posted by Glauber Costa on Mon, 29 Oct 2012 15:19:53 GMT
View Forum Message <> Reply to Message

On 10/29/2012 07:14 PM, JoonSoo Kim wrote:
> Hi, Glauber.
>
> 2012/10/19 Glauber Costa <glommer@parallels.com>:
>> We are able to match a cache allocation to a particular memcg. If the
>> task doesn't change groups during the allocation itself - a rare event,
>> this will give us a good picture about who is the first group to touch a
>> cache page.
>>
>> This patch uses the now available infrastructure by calling
>> memcg_kmem_get_cache() before all the cache allocations.
>>
>> 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>
>> CC: Tejun Heo <tj@kernel.org>
>> ---
>> include/linux/slub_def.h | 15 ++++++++++-----
>> mm/memcontrol.c | 3 +++
>> mm/slab.c | 6 +++++-
>> mm/slub.c | 5 +++--
>> 4 files changed, 21 insertions(+), 8 deletions(-)
>>
>> diff --git a/include/linux/slub_def.h b/include/linux/slub_def.h
>> index 961e72e..ed330df 100644
>> --- a/include/linux/slub_def.h
>> +++ b/include/linux/slub_def.h
>> @@ -13,6 +13,8 @@
>> #include <linux/kobject.h>
>>
>> #include <linux/kmemleak.h>
>> +#include <linux/memcontrol.h>
>> +#include <linux/mm.h>
>>
>> enum stat_item {
>> ALLOC_FASTPATH, /* Allocation from cpu slab */
>> @@ -209,14 +211,14 @@ static __always_inline int kmalloc_index(size_t size)
>> * This ought to end up with a global pointer to the right cache
>> * in kmalloc_caches.
>> */

```
>> -static __always_inline struct kmem_cache *kmalloc_slab(size_t size)
>> +static __always_inline struct kmem_cache *kmalloc_slab(gfp_t flags, size_t size)
>>  {
>>         int index = kmalloc_index(size);
>>
>>         if (index == 0)
>>                 return NULL;
>>
>> -       return kmalloc_caches[index];
>> +       return memcg_kmem_get_cache(kmalloc_caches[index], flags);
>>  }
>
> You don't need this,
> because memcg_kmem_get_cache() is invoked in both slab_alloc() and
> __cache_alloc_node().
>
```

Indeed, I had noticed this already, and fixed myself - to be sent in the
next version I intend to get out in the open tonight or tomorrow.