Subject: Re: [PATCH 04/11] kmem slab accounting basic infrastructure
Posted by Glauber Costa on Tue, 26 Jun 2012 07:09:24 GMT
View Forum Message <> Reply to Message

On 06/26/2012 08:22 AM, David Rientjes wrote:
> On Mon, 25 Jun 2012, Glauber Costa wrote:
>
>> diff --git a/mm/memcontrol.c b/mm/memcontrol.c
>> index 9352d40..6f34b77 100644
>> --- a/mm/memcontrol.c
>> +++ b/mm/memcontrol.c
>> @@ -265,6 +265,10 @@ struct mem_cgroup {
>> };
>>
>> /*
>> + * the counter to account for kernel memory usage.
>> + */
>> + struct res_counter kmem;
>> + /*
>> * Per cgroup active and inactive list, similar to the
>> * per zone LRU lists.
>> */
>> @@ -279,6 +283,7 @@ struct mem_cgroup {
>> * Should the accounting and control be hierarchical, per subtree?
>> */
>> bool use_hierarchy;
>> + bool kmem_accounted;
>>
>> bool oom_lock;
>> atomic_t under_oom;
>> @@ -391,6 +396,7 @@ enum res_type {
>> _MEM,
>> _MEMSWAP,
>> _OOM_TYPE,
>> + _KMEM,
>> };
>>
>> #define MEMFILE_PRIVATE(x, val) ((x) << 16 | (val))
>> @@ -1438,6 +1444,10 @@ done:
>> res_counter_read_u64(&memcg->memsw, RES_USAGE) >> 10,
>> res_counter_read_u64(&memcg->memsw, RES_LIMIT) >> 10,
>> res_counter_read_u64(&memcg->memsw, RES_FAILCNT));
>> + printk(KERN_INFO "kmem: usage %llukB, limit %llukB, failcnt %llu\n",
>> + res_counter_read_u64(&memcg->kmem, RES_USAGE) >> 10,
>> + res_counter_read_u64(&memcg->kmem, RES_LIMIT) >> 10,
>> + res_counter_read_u64(&memcg->kmem, RES_FAILCNT));
>> }
>>

>>   /*
>> @@ -3879,6 +3889,11 @@ static ssize_t mem_cgroup_read(struct cgroup *cont, struct cftype
*cft,
>>   else
>>   val = res_counter_read_u64(&memcg->memsw, name);
>>   break;
>> +#ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
>> + case _KMEM:
>> + val = res_counter_read_u64(&memcg->kmem, name);
>> + break;
>> +#endif
>
> This shouldn't need an #ifdef, ->kmem is available on all
> CONFIG_CGROUP_MEM_RES_CTLR kernels. Same with several of the other
> instances in this patch.
>
> Can't these instances be addressed by not adding kmem_cgroup_files without
> CONFIG_CGROUP_MEM_RES_CTLR_KMEM?

Yes, it can.
