Subject: Re: [PATCH 1/4] Add notification about some major slab events
Posted by Pavel Emelianov on Wed, 19 Sep 2007 10:08:32 GMT
View Forum Message <> Reply to Message

[snip]

>> @@ -1555,6 +1650,11 @@ static void __always_inline *slab_alloc(
>> }
>> local_irq_restore(flags);
>>
>> + if (object && slub_alloc_notify(s, object, gfpflags) < 0) {
>> + kmem_cache_free(s, object);
>> + return NULL;
>> + }
>> +
>> if (unlikely((gfpflags & __GFP_ZERO) && object))
>> memset(object, 0, c->objsize);
>>
>
> Please stay completely out of the fast path. No modifications to
> slab_alloc or slab_free please. It is possible to force all allocations of
> a particular slab of interest to use the slow path in __slab_alloc (maybe
> as a result of the slab page allocation hook returning a certain result
> code). See how the SLAB_DEBUG handling does it. You can adapt that and then do the
> object checks in __slab_alloc.

I have run the kernbench test on the kernels with a) containers support
and b) containers and kmem accounting support but (!) turned off. The
results are:

	a)	b)
Elapsed Time	768.500000	767.050000
User Time	679.360000	679.240000
System Time	87.020000	86.950000
Percent CPU	99.000000	99.000000
Context Switches	376891.000000	375407.000000
Sleeps	385377.000000	385426.000000

The test run was kernbench -n 1 -o 4 -M, the node is i386
DualCore Xeon 3.2GHz with 2Gb of RAM.

so the fast path is still fast, and we have two ways:
1. we keep the checks on the fastpath and have 0 overhead for
   unaccounted caches and some overhead for accounted;
2. we move the checks into the slow one and have 0 overhead for
   unaccounted caches and huge overhead for accounted.

I admit that I messed something, so shall I measure some
other activity or use another HW?

Thanks,
Pavel