Subject: Re: [PATCH 2/5] slab_id: Generic slab ID infrastructure
Posted by Pavel Emelyanov on Fri, 07 Oct 2011 08:31:51 GMT
View Forum Message <> Reply to Message

On 10/07/2011 12:27 PM, Glauber Costa wrote:
> Hi Pavel,
>
> On 10/06/2011 08:23 PM, Pavel Emelyanov wrote:
>> The idea of how to generate and ID for an arbitrary slab object is simple:
>>
>> - The ID is 128 bits
>> - The upper 64 bits are slab ID
>> - The lower 64 bits are object index withing a slab (yes, it's too many,
>> but is done for simplicity - not to deal with 96-bit numbers)
>> - The slab ID is the 48-bit per-cpu monotonic counter mixed with 16-bit
>> cpuid. Even if being incremented 1M times per second the first part
>> will stay uniqe for 200+ years. The cpuid is required to make values
>> picked on two cpus differ.
>
> So why can't we just use tighter numbers, and leave some reserved fields
> instead ?

Well, we have to save the ID on the slab and for 64-bit kernel we can already
use the 64-bit mapping field. For 32-bit kernels 32-bit value is not enough as
it can overlap in several days (like 32bit jiffies do) which is not enough.

> Having ids in the objects of the slab may prove useful in the future for
> other uses as well.
>
> For instance, concurrent to that, we're trying to figure out ways to
> have per-cgroup pages/objects accounted in the memory controller.
>
> The most up2date proposals create an entire kmem cache for each cgroup,
> thus trivially guaranteeing uniqueness. It however, leads to fragmentation.
> Having the objects to be IDed and being cgroup part of this id, could
> help us achieve the same goal with less fragmentation.

That's good point! I can extend the patches to provide the space reservation
infrastructure for slabs.

Thanks,
Pavel