Subject: Re: [PATCH 1/4] Add notification about some major slab events
Posted by Christoph Lameter on Tue, 18 Sep 2007 19:35:07 GMT
View Forum Message <> Reply to Message

On Tue, 18 Sep 2007, Pavel Emelyanov wrote:

> > 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.
>
> That's true, but:
> 1. we perform only a flag check on a fast path

This is still going to slow down everyone else and I still think there is
no need to do that.

> 2. currently we cannot force the freeing of an object to go _always_
> through the slow __slab_free(), and thus the following situation is
> possible:
> a. container A allocates an object and this object is
> accounted to it

At that point you could mark the slab as a slow slab by setting
SLAB_DEBUG() so we always take the slow path for this slab.

> b. the object is freed and gets into lockless freelist (but
> stays accounted to A)

B wont work if SLAB_DEBUG() is set. The fastpath is then disabled.

> c. container C allocates this object from the freelist
> and thus get unaccounted amount of memory

> this discrepancy can grow up infinitely. Sure, we can mark some caches to
> go through the slow path even on freeing the objects, but isn't it the
> same as checking for SLAB_NOTIFY on fast paths?

The other caches will then still perform up to speed.

> Maybe it's worth having the notifiers under config option, so that those
> who don't need this won't suffer at all?

I think you would want the container functionality to be available in
distros. They may make the choice whether to enable the container
functionality based on its impact. It is good if we can stash it away so

that there is a negligible performance impact if its compiled in but off.