Subject: Re: [PATCH v3 15/16] memcg/sl[au]b: shrink dead caches
Posted by Tejun Heo on Fri, 21 Sep 2012 20:40:35 GMT
View Forum Message <> Reply to Message

Hello, Glauber.

On Tue, Sep 18, 2012 at 06:12:09PM +0400, Glauber Costa wrote:
> @@ -764,10 +777,21 @@ static struct kmem_cache *memcg_create_kmem_cache(struct mem_cgroup *memcg,
>  goto out;
>  }
>
> + /*
> +  * Because the cache is expected to duplicate the string,
> +  * we must make sure it has opportunity to copy its full
> +  * name. Only now we can remove the dead part from it
> +  */
> + name = (char *)new_cachep->name;
> + if (name)
> +  name[strlen(name) - 4] = '\0';

This is kinda nasty. Do we really need to do this? How long would a dead cache stick around?

> diff --git a/mm/slab.c b/mm/slab.c
> index bd9928f..6cb4abf 100644
> --- a/mm/slab.c
> +++ b/mm/slab.c
> @@ -3785,6 +3785,8 @@ static inline void __cache_free(struct kmem_cache *cachep, void *objp,
>  }
>
>  ac_put_obj(cachep, ac, objp);
> +
> + kmem_cache_verify_dead(cachep);

Reaping dead caches doesn't exactly sound like a high priority thing and adding a branch to hot path for that might not be the best way to do it. Why not schedule an extremely lazy deferrable delayed_work which polls for emptiness, say, every miniute or whatever?

Thanks.

--
tejun