Subject: Re: [PATCH 3/7] per-cgroup slab caches
Posted by Suleiman Souhlal on Tue, 21 Feb 2012 23:50:32 GMT
View Forum Message <> Reply to Message

On Tue, Feb 21, 2012 at 3:34 AM, Glauber Costa <glommer@parallels.com> wrote:
> diff --git a/mm/memcontrol.c b/mm/memcontrol.c
> index 26fda11..2aa35b0 100644
> --- a/mm/memcontrol.c
> +++ b/mm/memcontrol.c
> +struct kmem_cache *
> +kmem_cache_dup(struct mem_cgroup *memcg, struct kmem_cache *base)
> +{
> +	struct kmem_cache *s;
> +	unsigned long pages;
> +	struct res_counter *fail;
> +	/*
> +	 * TODO: We should use an ida-like index here, instead
> +	 * of the kernel address
> +	 */
> +	char *kname = kasprintf(GFP_KERNEL, "%s-%p", base->name, memcg);

Would it make more sense to use the memcg name instead of the pointer?

> +
> +	WARN_ON(mem_cgroup_is_root(memcg));
> +
> +	if (!kname)
> +		return NULL;
> +
> +	s = kmem_cache_create_cg(memcg, kname, base->size,
> +				 base->align, base->flags, base->ctor);
> +	if (WARN_ON(!s))
> +		goto out;
> +
> +
> +	pages = slab_nr_pages(s);
> +
> +	if (res_counter_charge(memcg_kmem(memcg), pages << PAGE_SHIFT, &fail)) {
> +		kmem_cache_destroy(s);
> +		s = NULL;
> +	}

What are we charging here? Does it ever get uncharged?

-- Suleiman