Subject: Re: [RFC][-mm PATCH 4/8] Memory controller memory accounting (v3)
Posted by Paul Menage on Fri, 20 Jul 2007 21:03:36 GMT
View Forum Message <> Reply to Message

On 7/20/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

> +void __always_inline unlock_meta_page(struct page *page)
> +{
> +	bit_spin_unlock(PG_metapage, &page->flags);
> +}

Maybe add a BUG_ON(!test_bit(PG_metapage, &page->flags)) at least for development?

> +	mem = rcu_dereference(mm->mem_container);
> +	/*
> +	 * For every charge from the container, increment reference
> +	 * count
> +	 */
> +	css_get(&mem->css);
> +	rcu_read_unlock();

It's not clear to me that this is safe.

If

> +
> +	/*
> +	 * If we created the meta_page, we should free it on exceeding
> +	 * the container limit.
> +	 */
> +	if (res_counter_charge(&mem->res, 1)) {
> +		css_put(&mem->css);
> +		goto free_mp;
> +	}
> +
> +	lock_meta_page(page);
> +	/*
> +	 * Check if somebody else beat us to allocating the meta_page
> +	 */
> +	if (page_get_meta_page(page)) {

I think you need to add something like

  kfree(mp);
  mp = page_get_meta_page(page);

otherwise you're going to leak the new but unneeded metapage.

```
> +               atomic_inc(&mp->ref_cnt);
> +               res_counter_uncharge(&mem->res, 1);
> +               goto done;
> +       }
> +
> +       atomic_set(&mp->ref_cnt, 1);
> +       mp->mem_container = mem;
> +       mp->page = page;
> +       page_assign_meta_page(page, mp);
```

Would it make sense to have the "mp->page = page" be part of page_assign_meta_page() for consistency?

```
> +err:
> +       unlock_meta_page(page);
> +       return -ENOMEM;
```

The only jump to err: is from a location where the metapage is already unlocked. Maybe scrap err: and just do a return -ENOMEM when the allocation fails?

```
> +out_uncharge:
> +       mem_container_uncharge(page_get_meta_page(page));
```

Wanting to call mem_container_uncharge() on a page and hence having to call page_get_meta_page() seems to be more common than wanting to call it on a meta page that you already have available. Maybe make mem_container_uncharge() be a wrapper that take a struct page and does something like mem_container_uncharge_mp(page_get_meta_page(page)) where mem_container_uncharge_mp() is the raw meta-page version?

Paul