
Subject: Re: [PATCH v8 1/9] Basic kernel memory functionality for the Memory Controller

Posted by [KAMEZAWA Hiroyuki](#) on Fri, 09 Dec 2011 01:21:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Mon, 5 Dec 2011 19:34:55 -0200

Glauber Costa <glommer@parallels.com> wrote:

> This patch lays down the foundation for the kernel memory component
> of the Memory Controller.

>

> As of today, I am only laying down the following files:

>

> * memory.independent_kmem_limit
> * memory.kmem.limit_in_bytes (currently ignored)
> * memory.kmem.usage_in_bytes (always zero)

>

> Signed-off-by: Glauber Costa <glommer@parallels.com>

> Reviewed-by: Kirill A. Shutemov <kirill@shutemov.name>

> CC: Paul Menage <paul@paulmenage.org>

> CC: Greg Thelen <gthelen@google.com>

As I wrote, please CC Johannes and Michal Hocko for memcg related parts.

A few questions.

==

> + val = !!val;
> +
> + if (parent && parent->use_hierarchy &&
> + (val != parent->kmem_independent_accounting))
> + return -EINVAL;
==

Hm, why you check val != parent->kmem_independent_accounting ?

```
if (parent && parent->use_hierarchy)
    return -EINVAL;
?
```

BTW, you didn't check this cgroup has children or not.

I think

```
if (this_cgroup->use_hierarchy &&
    !list_empty(this_cgroup->children))
    return -EINVAL;
```

==

> + /*

```
> + * TODO: We need to handle the case in which we are doing
> + * independent kmem accounting as authorized by our parent,
> + * but then our parent changes its parameter.
> + */
> + cgroup_lock();
> + memcg->kmem_independent_accounting = val;
> + cgroup_unlock();
```

Do we need cgroup_lock() here ?

Thanks,
-Kame
