Subject: Re: [RFC] memory controller : backgorund reclaim and avoid excessive locking [5/5] lazy page_cgroup
Posted by Balbir Singh on Mon, 18 Feb 2008 04:35:16 GMT
View Forum Message <> Reply to Message

KAMEZAWA Hiroyuki wrote:

```
> A big lock contetion of memory controller is mz->lru_lock.
>
> This is acquired when
>  1. add to lru
>  2. remove from lru
>  3. scan lru list
>
> It seems 1. and 3. are unavoidable. but 2. can be delayed.
>
> This patch make removing page_cgorup from lru-list be lazy and batched.
> (Like pagevec..)
>
> This patch adds a new flag page_cgroup and make lru scan routine
> ignores it.
>
>
> I think this reduces lock contention especially when
>  - several tasks are exiting at once.
>  - several files are removed at once.
>
>
> Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
>
```

Hi, KAMEZAWA-San,

For this and the next patch, do you know if there is a performance improvement? Have we measured it. The complexity of the code seems to be high, want to make sure it is worth it.

Could we re-use the pagevec mechansim for LRU handling? That was my long term plan and I also wanted to move the kmalloc's to their own cache and use kmem_cache_alloc and try some other experiments. May be batching the alloc's for page_container can be done in the same way that you've proposed LRU and accounting changes.

--
 Warm Regards,
 Balbir Singh
 Linux Technology Center
 IBM, ISTL