
Subject: Re: [RFC][only for review] memory controller background reclaim [2/5]
set/get ops for res_counter

Posted by [Pavel Emelianov](#) on Wed, 28 Nov 2007 11:09:26 GMT

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

KAMEZAWA Hiroyuki wrote:

```
> At implementing high/low watermark in res_counter, it will be better to
> adjust high/low value when limit changes. (or don't allow user to specify
> high/low value)
>
> This patch adds *internal* interface to modify resource value.
> (If there are only limit/usage/failcnt, these routines are not necessary but..)
> And will be used later.
>
> Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
>
> include/linux/res_counter.h | 7 ++++++
> kernel/res_counter.c       | 19 ++++++
> 2 files changed, 26 insertions(+)
>
> Index: linux-2.6.24-rc3-mm1/include/linux/res_counter.h
> =====
> --- linux-2.6.24-rc3-mm1.orig/include/linux/res_counter.h 2007-11-28 14:18:21.000000000
+0900
> +++ linux-2.6.24-rc3-mm1/include/linux/res_counter.h 2007-11-28 14:18:33.000000000 +0900
> @@ -59,6 +59,13 @@
>  int (*write_strategy)(char *buf, unsigned long long *val));
>
> /*
>  * A routine for set/get limitation value from kernel internal code.
>  * res->lock should be held before call this.
>  */
> +unsigned long long res_counter_get(struct res_counter *counter, int member);
> +void res_counter_set(struct res_counter *conter, int member,
> + unsigned long long val);
> +/*
>  * the field descriptors. one for each member of res_counter
>  */
>
> Index: linux-2.6.24-rc3-mm1/kernel/res_counter.c
> =====
> --- linux-2.6.24-rc3-mm1.orig/kernel/res_counter.c 2007-11-28 14:18:21.000000000 +0900
> +++ linux-2.6.24-rc3-mm1/kernel/res_counter.c 2007-11-28 14:18:33.000000000 +0900
> @@ -75,6 +75,25 @@
>  return NULL;
> }
>
> +unsigned long long res_counter_get(struct res_counter *res, int member)
```

```
> +{  
> + unsigned long long *val;  
> +  
> + val = res_counter_member(res, member);  
> +  
> + return *val;  
> +}  
> +  
> +void res_counter_set(struct res_counter *res, int member,  
> + unsigned long long newval)  
> +{  
> + unsigned long long *val;  
> +  
> + val = res_counter_member(res, member);  
> + *val = newval;
```

You put locking here in the `res_counter_write()` (patch #1). Why is it missed here?

```
> + return;  
> +}  
> +  
> ssize_t res_counter_read(struct res_counter *counter, int member,  
> const char __user *userbuf, size_t nbytes, loff_t *pos,  
> int (*read_strategy)(unsigned long long val, char *st_buf))  
>  
>
```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
