
Subject: Re: [PATCH 1/2] Add the max_usage member on the res_counter

Posted by [Balbir Singh](#) on Sat, 08 Mar 2008 05:39:14 GMT

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

Pavel Emelyanov wrote:

```
> This is a very usefull feature. E.g. one may set the
> limit to "unlimited" value and check for the memory
> requirements of a new container.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> ---
> include/linux/res_counter.h |  5 +++++
> kernel/res_counter.c       |  4 +++
> mm/memcontrol.c           |  5 +////
> 3 files changed, 14 insertions(+), 0 deletions(-)
>
> diff --git a/include/linux/res_counter.h b/include/linux/res_counter.h
> index 8cb1ecd..2c4deb5 100644
> --- a/include/linux/res_counter.h
> +++ b/include/linux/res_counter.h
> @@ -25,6 +25,10 @@ struct res_counter {
>   */
>   unsigned long long usage;
>   /*
> + * the maximal value of the usage from the counter creation
> + */
> + unsigned long long max_usage;
> + /*
>   * the limit that usage cannot exceed
>   */
>   unsigned long long limit;
> @@ -67,6 +71,7 @@ ssize_t res_counter_write(struct res_counter *counter, int member,
>
> enum {
>   RES_USAGE,
> + RES_MAX_USAGE,
>   RES_LIMIT,
>   RES_FAILCNT,
> };
> diff --git a/kernel/res_counter.c b/kernel/res_counter.c
> index 791ff2b..f1f20c2 100644
> --- a/kernel/res_counter.c
> +++ b/kernel/res_counter.c
> @@ -27,6 +27,8 @@ int res_counter_charge_locked(struct res_counter *counter, unsigned
long val)
> }
>
```

```
> counter->usage += val;  
> + if (counter->usage > counter->max_usage)  
> + counter->max_usage = counter->usage;
```

How about

```
counter->max_usage = max(counter->usage, counter->max_usage);
```

```
> return 0;  
> }  
>  
> @@ -65,6 +67,8 @@ res_counter_member(struct res_counter *counter, int member)  
> switch (member) {  
> case RES_USAGE:  
>     return &counter->usage;  
> + case RES_MAX_USAGE:  
> +     return &counter->max_usage;  
> case RES_LIMIT:  
>     return &counter->limit;  
> case RES_FAILCNT:  
> diff --git a/mm/memcontrol.c b/mm/memcontrol.c  
> index 2d59163..e5c741a 100644  
> --- a/mm/memcontrol.c  
> +++ b/mm/memcontrol.c  
> @@ -911,6 +911,11 @@ static struct cftype mem_cgroup_files[] = {  
>     .read_u64 = mem_cgroup_read,  
> },  
> {  
> + .name = "max_usage_in_bytes",  
> + .private = RES_MAX_USAGE,  
> + .read_u64 = mem_cgroup_read,  
> + },  
> + {  
>     .name = "limit_in_bytes",  
>     .private = RES_LIMIT,  
>     .write = mem_cgroup_write,
```

Looks very good,

Acked-by: Balbir Singh <balbir@linux.vnet.ibm.com>

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

Containers mailing list

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