

---

Subject: Re: [PATCH] Simplify memory controller and resource counter I/O  
Posted by [Balbir Singh](#) on Fri, 05 Oct 2007 03:31:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Paul Menage wrote:

> Hi Balbir,  
>  
> Any thoughts on this patch?  
>

Hi, Paul,

I remember seeing this patch, sorry for not responding earlier

> Cheers,  
>  
> Paul  
>  
> On 9/25/07, Paul Menage <[menage@google.com](mailto:menage@google.com)> wrote:  
>> Simplify the memory controller and resource counter I/O routines  
>>  
>> This patch strips out some I/O boilerplate from resource counters and  
>> the memory controller. It also adds locking to the resource counter  
>> reads and writes, and forbids writes to the root memory cgroup's limit  
>> file.  
>>

Forbidding writing to the root resource counter is a policy decision  
I am unable to make up my mind about. It sounds right, but unless  
we have a notion of unlimited resources, I am a bit concerned about  
taking away this flexibility.

>> One arguable drawback to this patch is that the use of `memparse()` is  
>> lost in the cleanup. Having said that, given the existing of shell  
>> arithmetic, it's not clear to me that typing  
>>

`memparse()`, makes it so much easier, we need to use it.

>> `echo ${2<<30} > memory.limit`  
>>

Very geeky! I don't like it personally

>> is especially harder than  
>>  
>> `echo 2G > memory.limit`  
>>

>> Signed-off-by: Paul Menage <menage@google.com>  
>>

I like the read\_uint() and write\_uint() overall, but in the case of setting the limit, I'd still like the flexibility of having a strategy pattern that would make the UI more friendly.

Do read\_uint() and write\_uint(), just read and write unsigned integers?

[snip]

--

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>

---