

---

Subject: Re: [RFC][-mm PATCH 2/8] Memory controller containers setup (v3)  
Posted by [Paul Menage](#) on Fri, 20 Jul 2007 20:30:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On 7/20/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

```
>
> +config CONTAINER_MEM_CONT
> +    bool "Memory controller for containers"
> +    select CONTAINERS
```

Andrew asked me to not use "select" in Kconfig files due to some unspecified problems seen in the past, so my latest patchset makes the subsystems depend on containers rather than selecting them; I prefer the select approach over the dependency approach, but if select does have problems then we should be consistent.

```
> +static struct cftype mem_container_usage = {
> +    .name = "mem_usage",
> +    .private = RES_USAGE,
```

For V11, the .name field should just be called something like 'usage'; the subsystem name is automatically prefixed.

```
> +
> +static int mem_container_create(struct container_subsys *ss,
> +                               struct container *cont)
> +{
> +    struct mem_container *mem;
> +
> +    mem = kzalloc(sizeof(struct mem_container), GFP_KERNEL);
> +    if (!mem)
> +        return -ENOMEM;
> +
> +    res_counter_init(&mem->res);
> +    cont->subsys[mem_container_subsys_id] = &mem->css;
> +    mem->css.container = cont;
> +    return 0;
```

For the V11 patchset, you'll want to replace these three lines with just

```
    return &mem->css;
```

```
> +static int mem_container_populate(struct container_subsys *ss,
> +                               struct container *cont)
> +{
> +    int rc = 0;
> +
> +    rc = container_add_file(cont, &mem_container_usage);
```

```

> +    if (rc < 0)
> +        goto err;
> +
> +    rc = container_add_file(cont, &mem_container_limit);
> +    if (rc < 0)
> +        goto err;
> +
> +    rc = container_add_file(cont, &mem_container_failcnt);
> +    if (rc < 0)
> +        goto err;

```

There's a `container_add_files()` API in V10 and above that lets you register an array of files in one go.

```

> +
> +err:
> +    return rc;
> +}
> +
> +struct container_subsys mem_container_subsys = {
> +    .name = "mem_container",

```

Maybe just "memory" or "pages" for the container name?

Paul

---

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

---



---

Subject: Re: [RFC][-mm PATCH 2/8] Memory controller containers setup (v3)  
Posted by [Balbir Singh](#) on Sat, 21 Jul 2007 17:04:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Paul Menage wrote:

> On 7/20/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

>>

>> +config CONTAINER\_MEM\_CONT

>> + bool "Memory controller for containers"

>> + select CONTAINERS

>

> Andrew asked me to not use "select" in Kconfig files due to some  
> unspecified problems seen in the past, so my latest patchset makes the  
> subsystems depend on containers rather than selecting them; I prefer  
> the select approach over the dependency approach, but if select does  
> have problems then we should be consistent.

>

I'll change it to depends

```
>> +static struct cftype mem_container_usage = {
>> +    .name = "mem_usage",
>> +    .private = RES_USAGE,
>
> For V11, the .name field should just be called something like 'usage';
> the subsystem name is automatically prefixed.
>
```

Will change

```
>> +
>> +static int mem_container_create(struct container_subsys *ss,
>> +    struct container *cont)
>> +{
>> +    struct mem_container *mem;
>> +
>> +    mem = kzalloc(sizeof(struct mem_container), GFP_KERNEL);
>> +    if (!mem)
>> +        return -ENOMEM;
>> +
>> +    res_counter_init(&mem->res);
>> +    cont->subsys[mem_container_subsys_id] = &mem->css;
>> +    mem->css.container = cont;
>> +    return 0;
>
> For the V11 patchset, you'll want to replace these three lines with just
>
> return &mem->css;
>
```

Will do

```
>> +static int mem_container_populate(struct container_subsys *ss,
>> +    struct container *cont)
>> +{
>> +    int rc = 0;
>> +
>> +    rc = container_add_file(cont, &mem_container_usage);
>> +    if (rc < 0)
>> +        goto err;
>> +
>> +    rc = container_add_file(cont, &mem_container_limit);
>> +    if (rc < 0)
>> +        goto err;
>> +
```

```
>> + rc = container_add_file(cont, &mem_container_failcnt);
>> + if (rc < 0)
>> +     goto err;
>
> There's a container_add_files() API in V10 and above that lets you
> register an array of files in one go.
>
```

Cool! I'll migrate to that.

```
>> +
>> +err:
>> +     return rc;
>> +}
>> +
>> +struct container_subsys mem_container_subsys = {
>> +    .name = "mem_container",
>
> Maybe just "memory" or "pages" for the container name?
>
```

Traditionally names like memory\_control have been used to indicate the purpose of the controller. I understand that container is an overloaded term, so I guess memory might be a better name.

> Paul

--

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>

---