Subject: Re: [PATCH 4/8] RSS container core
Posted by Pavel Emelianov on Thu, 31 May 2007 08:57:26 GMT

View Forum Message <> Reply to Message

Andrew Morton wrote:

```
>> +#else
>> +static inline int container_rss_prepare(struct page *pg,
>> +  struct vm_area_struct *vma, struct page_container **pc)
>> +{
>> + *pc = NULL; /* to make gcc happy */
>
> eh?  What is gcc's problem here?
```

When this line is missed and CONFIG_RSS_COUNTER is n the following warnings are produced:

```
  CC      mm/memory.o
```

```
>> + return 0;
>> +}
>>
>> ...
>>
>> +static int rss_populate(struct container_subsys *ss,
>> +  struct container *cont)
>> +{
>> + int rc;
>> +
>> + if ((rc = container_add_file(cont, &rss_usage)) < 0)
>> +  return rc;
>> + if ((rc = container_add_file(cont, &rss_failcnt)) < 0)
>> +  return rc;
>> + if ((rc = container_add_file(cont, &rss_limit)) < 0)
>> +  return rc;
>> + if ((rc = container_add_file(cont, &rss_reclaimed)) < 0)
>> +  return rc;
>
> If we fail partway through here, do the thus-far-created fiels get cleaned up?
```

Yes. As far as I see from Paul's code when one of the files is failed to
be created the whole container is cleaned up.

```
>> + return 0;
>> +}
>> +
>> +struct container_subsys rss_subsys = {
>> + .name = "rss",
>> + .subsys_id = rss_subsys_id,
>> + .create = rss_create,
>> + .destroy = rss_destroy,
>> + .populate = rss_populate,
>> + .attach = rss_move_task,
>> + .early_init = 1,
>> +};
>
> Did this need kernel-wide scope?
```

Yes. In include/linux/countainer_subsys.h we declared the SUBSYS(rss)
that is expanded into "extern struct container_subsys rss_subsys;" in the
kernel/container.c file. Further the pointer to it is injected into the array
of subsystems (static struct container_subsys *subsys[]; at the top of the file).