
Subject: Re: containers development plans (July 10 version)

Posted by [Balbir Singh](#) on Wed, 11 Jul 2007 07:21:17 GMT

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

Paul Menage wrote:

> On 7/10/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

>>

>> Well, it depends on how you define ugly. We could do something like

>> the namespace approach, have something like

>>

>> struct mem_container_ptrs {

>> swap_list;

>> mem_container_ptr;

>> };

>

> I'm not quite sure what you're aiming for there. What would swap_list

> represent?

>

swap_list is a list of swap_devices associated with the container.

the mem_container_ptr points to the mem_container which in turn

knows which container it belongs to.

> I'm wondering if for both the per-page controller and the swap

> controller, it would make sense to have a pointer back to an

> appropriate process so we could get at a container pointer

>

> Maybe something like:

>

> - when an mm is created, store a pointer to the task_struct that it

> belongs to

> - when a process exits and its mm_struct points to it, and there are

> other mm users (i.e. a thread group leader exits before some of its

> children), then find a different process that's using the same mm

> (which will almost always be the next process in the list running

> through current->tasks, but in strange situations we might need to

> scan the global tasklist)

>

We'll that sounds like a complicated scheme.

> Then rather than having to have a pointer in the mm for either the

> page controller or the swap controller (and the consequent hassles of

> having refcounts from mm_structs to containers), you can just use the

> container membership of mm->owner.

>

We do that currently, our mm->owner is called mm->mem_container. It points

to a data structure that contains information about the container to which the mm belongs. The problem I see with mm->owner is that several threads can belong to different containers. I see that we probably mean the same thing, except that you suggest using a pointer to the task_struct from mm_struct, which I am against in principle, due to the complexity of changing owners frequently if the number of threads keep exiting at a rapid rate.

>>
>> Although, I agree that per container swap is important, I feel that
>> we should add in the functionality, once we have basic page based
>> memory controller. It would make the whole setup easier to test
>> for functionality and performance.

>
> We don't really need to wait for a working page-based memory
> controller to be able to test a swap controller - cpusets gives memory
> controls too, albeit on a much coarser granularity.
>

We have a working page-based memory controller, it is yet to find it's way into -mm though. The implementation of per container swap would be useful and should work for both nevertheless. For cpusets to provide memory control on non NUMA machines, we need to get in fuke numa emulation support into all architectures.

> Paul

--

Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL
