Subject: Re: [ckrm-tech] containers development plans (July 10 version)
Posted by Paul Menage on Wed, 11 Jul 2007 12:18:52 GMT
View Forum Message <> Reply to Message

On 7/11/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
> Paul Menage wrote:
> > On 7/11/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
> >> swap_list is a list of swap_devices associated with the container.
> >
> > That doesn't sound so great, since you'd need to update all the
> > mem_container_ptr objects that point to that swap controller subsys
> > state when you change the swap devices for the container.
> >
>
> Not all of them, only for that container. This list is per container.
> I don't see why need to update all the mem_container_ptr objects?

What if the mm is in different containers for the swap controller and the page controller? (i.e. the two controllers were mounted on different hierarchies, which can easily be the case if one of them is in use, and the other isn't).

In that case you'd end up with one mem_container_ptr object for each combination of (swap container, page container) and would basically be reimplementing the css_group support, but for mm_struct rather than task_struct.

And since there could be multiple mem_container_ptr objects for the same swap controller container state, you'd need to update multiple lists.

> >
> > I don't think it's that complicated. There would be some slightly
> > interesting synchronization, probably involving RCU, to make sure you
> > didn't derefence mm->owner when mm->owner had been freed but apart
> > from that it's straightforward.
> >
>
> Walking the global tasklist to find the tasks that share the same mm
> to me seems like an overhead.

As I mentioned above, I think that would be very rare, because:

1) Most tasks when they exit would either not be the mm owner (child threads) or else there would be no other mm users (non-threaded apps)

2) If you're the mm owner and there are other users, the most likely place to find another user of that mm would be to find the next

task_struct in your task group.

3) If there are no other tasks in your task group then one of your
siblings or children will probably be one of the other users

4) In very rare cases (not sure any come to mind right now, but maybe
if you were doing funky things with clone) you might need to walk the
global tasklist.

>
> Hmmm.. interesting.. I was there, but I guess I missed the discussion
> (did u have it after the talk?)

It was one of the questions that Pavel asked. He was asking in the
context of processes changing container, and having resources left
behind in the old container. It's basically the problem of when you
consume non-renewable resources that aren't uniquely associated with a
single task struct, what happens if some of the tasks that are
responsible get moved to a different container. With a unique owner
task for each mm, at least we wouldn't face this problem with
mm_struct any longer, although the individual pages still have this
problem.

Paul