Subject: Re: [ckrm-tech] [PATCH 3/6] containers: Add generic multi-subsystem API to containers
Posted by Balbir Singh on Fri, 12 Jan 2007 06:29:52 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 1/10/07, Balbir Singh <balbir@in.ibm.com> wrote:
>> Paul Menage wrote:
>>> +/* The set of hierarchies in use. Hierarchy 0 is the "dummy
>>> + * container", reserved for the subsystems that are otherwise
>>> + * unattached - it never has more than a single container, and all
>>> + * tasks are part of that container. */
>>> +
>>> +static struct containerfs_root rootnode[CONFIG_MAX_CONTAINER_HIERARCHIES];
>>> +
>>> +/* dummytop is a shorthand for the dummy hierarchy's top container */
>>> +#define dummytop (&rootnode[0].top_container)
>>> +
>> With these changes, is there a generic way to determine the root container
>> for the hierarchy the subsystem is in? Calls to ->create() pass the dummytop
>> container.
>
> There are two places that the subsystem create() function is called -
> the first is during the subsystem registration, to create the
> subsystem state for the root container. That one passes in dummytop
> since that is the container that all subsystems start attached to.
>

Yes, I saw that.

> For clarification, the default (dummy) hierarchy is a placeholder for
> subsystems that aren't bound to a hierarchy. It always contains
> exactly one container (dummytop) and all processes are members of that
> container. It isn't reference-counted, since it can never go away, and
> it can never have any subcontainers.
>
> When a real subcontainer is created (which must be after a subsystem
> has been bound to a hierarchy via a filesystem mount), the new
> subcontainer is passed in. From there you can follow the top_container
> field in the subcontainer, which leads to the root of the hierarchy.
>
> Andrew has suggested that I need to document this better :-)
>

One of things I was trying to do with cpu_acct was to actually calculate
the % load over a defined interval. I have the patch for that ready.
When the interval ticks over (which happens in interrupt context -
account_xxxxx_time()), I want to reset the load of child containers

to 0. To walk the hierarchy, I have no root now since I do not have
any task context. I was wondering if exporting the rootnode or providing
a function to export the rootnode of the mounter hierarchy will make
programming easier.

Something like

```
struct container *get_root_container(struct container_subsys *ss)
{
 return &rootnode[ss->hierarchy];
}
```

> Paul
>

--

Balbir Singh,
Linux Technology Center,
IBM Software Labs