Subject: Re: [ckrm-tech] [PATCH 3/9] Containers (V9): Add tasks file interface
Posted by Balbir Singh on Fri, 11 May 2007 02:31:30 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 5/8/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>>
>> I now have a use case for maintaining a per-container task list.
>> I am trying to build a per-container stats similar to taskstats.
>> I intend to support container accounting of
>>
>> 1. Tasks running
>> 2. Tasks stopped
>> 3. Tasks un-interruptible
>> 4. Tasks blocked on IO
>> 5. Tasks sleeping
>>
>> This would provide statistics similar to the patch that Pavel had sent
>> out.
>>
>> I faced the following problems while trying to implement this feature
>>
>> 1. There is no easy way to get a list of all tasks belonging to a
>> container
>>   (we need to walk all threads)
>
> Well, walking the taks list is pretty easy - but yes, it could become
> inefficient when there are many small containers in use.
>
> I've got some ideas for a way of tracking this specifically for
> containers with subsystems that want this, while avoiding the overhead
> for subsystems that don't really need it. I'll try to add them to the
> next patchset.

Super!

>
>> 2. There is no concept of a container identifier. When a user issues a
>> command
>>   to extract statistics, the only unique container identifier is the
>> container
>>   path, which means that we need to do a path lookup to determine the
>> dentry
>>   for the container (which gets quite ugly with all the string
>> manipulation)
>
> We could just cache the container path permanently in the container,
> and invalidate it if any of its parents gets renamed. (I imagine this

> happens almost never.)
>

Here's what I have so far, I cache the mount point of the container
and add the container path to it. I'm now stuck examining tasks,
while walking through a bunch of tasks, there is no easy way of
knowing the container path of the task without walking all subsystems
and then extracting the containers absolute path.

>>
>>   Adding a container id, will make it easier to find a container and
>> return
>>   statistics belonging to the container.
>
> Not unreasonable, but there are a few questions that would have to be
> answered:
>
> - how is the container id picked? Like a pid, or user-defined? Or some
> kind of string?
>

I was planning on using a hierarchical scheme, top 8 bits for
the container hierarchy and bottom 24 for a unique id. The id
is automatically selected. Once we know the container id, we'll
need a more efficient mechanism to map the id to the container.

> - how would it be exposed to userspace? A generic control file
> provided by the container filesystem in all container directories?
>

A file in all container directories is an option

> - can you give a more concrete example of how this would actually be
> useful? For your container stats, it seems that just reading a control
> file in the container's directory would give you the stats that you
> want, and userspace already knows the container's name/id since it
> opened the control file.
>

Sure, the plan is to build a containerstats interface like taskstats.
In taskstats, we exchange data between user space and kernel space
using genetlink sockets. We have a push and pull mechanism for statistics.

> Paul

--

Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL