
Subject: Re: [PATCH 1/4] Virtualization/containers: introduction
Posted by [ebiederm](#) on Wed, 08 Feb 2006 05:03:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hubertus Franke <frankeh@watson.ibm.com> writes:

>
> Agreed.. here are some issues we learned from other projects that had
> similar interception points.
>
> Having a central umbrella object (let's stick to the name container)
> is useful, but being the only object through which every access has to
> pass may have drawbacks..
>
> task->container->pspace->pidmap[offset].page implies potential
> cachemisses etc.
>
> If overhead becomes too large, then we can stick (cache) the pointer
> additionally in the task struct. But ofcourse that should be carefully
> examined on a per subsystem base...

Ok. After talking with the vserver guys on IRC. I think grasp the importance. The key feature is to have a place to put limits and the like for your entire container. Look at all of the non-signal stuff in struct signal for an example. The nested namespaces seem to be just an implementation detail.

For OpenVZ having the other namespaces nested may have some importance. I haven't gotten their yet.

The task->container->pspace->.... thing feels very awkward to me, and feels like it increases our chance getting a cache miss.

So I support the concept of a place to put all of the odd little things like rlimits for containers. But I would like to flatten it in the task_struct if we can.

Eric
