Subject: Re: [RFC PATCH 0/5] Resend -v2 - Use procfs to change a syscall behavior
Posted by ebiederm on Fri, 18 Jul 2008 02:40:45 GMT
View Forum Message <> Reply to Message

Oren Laadan <orenl@cs.columbia.edu> writes:

> I think the kernel space vs. user space must be the first issue on our
> table to solve, as it has a wide impact on the rest of the work.

We first need to talk about what kinds of problems we are trying to solve. If we don't agree what the problem is I expect we will have a hard time agreeing on a solution.

For example we are using namespaces now instead of the potentially simpler isolation mechanism of Vserver because checkpoint/restart could not be done with the Vserver approach.

The use case that I expect we all have in common is migrating an isolated container from one machine to another transparent to applications. Except those that directly access the hardware at which point we can treat it as a hotplug event from the perspective of userspace.

There are several other interesting use cases that I think we should solve if possible.
- Live/Incremental migration.
- Remote fork. Which can be seen as an extreme case of migrating
  only a partial container.
- A checkpoint that can be restarted multiple times and work properly.
  Which means you need to include the state of the filesystem.
- A distributed checkpoint of multiple containers at the same time.

Given how brutally hard and inefficient it is to restore a checkpoint using the existing system calls even with namespaces in the kernel. We can pretty much rule that implementation out as it does not match our efficiency criteria, and likely isn't especially maintainble either.

On the maintenance side we can generally rule out an out of tree module. As that does not afford visible to people changing a subsystem that the checkpoint/restart code needs to change as well.

I believe the live migration will have the most stringent performance requirements and at the same time be one of the most useful features, as it immediately improve maintenance of clusters.

In the extreme case of a distributed checkpoint the kernel simply does not have enough state so we need user space code coordinating all of the pieces.

For a multi-start checkpoint I expect userspace will be coordinating
filesystem snapshots and checkpoints.

Eric