Subject: RE: [PATCH 0/9] namespaces: Introduction
Posted by Hua Zhong on Fri, 19 May 2006 18:28:08 GMT
View Forum Message <> Reply to Message

> snapshot/restart/migration worry me. If they require
> complete serialisation of complex kernel data structures then
> we have a problem, because it means that any time anyone
> changes such a structure they need to update (and test) the
> serialisation.

Checkpoint/Restart/Migration could be very complicated if done at OS level (per process/process group/or any subset of an OS). But
it is much simpler if done on virtual machine level (VMWare/Xen) because there is a natural and clear boundary, and doesn't get
affected if the OS kernel internal changes.

It's good to see some progress in supporting virtualization in Linux, but as Andrew put it, some big decisions need to be made
up-front. One big question is actually how many virtualization technologies Linux should support? Particularly, does it need to
support both OS-level virtualization and VM-level virtualization? And why? And to what degree?

My gut feeling is that the VM approach seems much cleaner and modular, without touching too many areas (except some low-level ones)
inside the kernel and in general very well separated. There are two reasons:

1. In a VM system, the architecture is very simple. Hypervisor and guest OS kernel have clear boundaries and interfaces to
communicate, and OS in general pretty much doesn't need to care if it's running on native hardware or inside a VM. So it adds very
little maintanence burden to the kernel developers (and if there is, it's relatively well understood).

2. Hardware support. With more virtualization built into CPU, VM is only going to get simper.

It seems at least the VM approach is much less risky. It might be helpful if someone could explain why we need both.

Hua