Subject: Re: [patch00/05]: Containers(V2)- Introduction
Posted by Peter Zijlstra on Wed, 20 Sep 2006 18:27:04 GMT
View Forum Message <> Reply to Message

On Wed, 2006-09-20 at 11:14 -0700, Rohit Seth wrote:
> On Wed, 2006-09-20 at 20:06 +0200, Peter Zijlstra wrote:
> > On Wed, 2006-09-20 at 10:52 -0700, Christoph Lameter wrote:
> > > On Wed, 20 Sep 2006, Rohit Seth wrote:
> > >
> > > > Right now the memory handler in this container subsystem is written in
> > > > such a way that when existing kernel reclaimer kicks in, it will first
> > > > operate on those (container with pages over the limit) pages first. But
> > > > in general I like the notion of containerizing the whole reclaim code.
> > >
> > > Which comes naturally with cpusets.
> >
> > How are shared mappings dealt with, are pages charged to the set that
> > first faults them in?
> >
>
> For anonymous pages (simpler case), they get charged to the faulting
> task's container.
>
> For filesystem pages (could be shared across tasks running different
> containers): Every time a new file mapping is created, it is bound to a
> container of the process creating that mapping. All subsequent pages
> belonging to this mapping will belong to this container, irrespective of
> different tasks running in different containers accessing these pages.
> Currently, I've not implemented a mechanism to allow a file to be
> specifically moved into or out of container. But when that gets
> implemented then all pages belonging to a mapping will also move out of
> container (or into a new container).

Yes, I read that in your patches, I was wondering how the cpuset
approach would handle this.

Neither are really satisfactory for shared mappings.