
Subject: Re: [patch00/05]: Containers(V2)- Introduction
Posted by [Nick Piggin](#) on Wed, 20 Sep 2006 16:44:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, Sep 20, 2006 at 09:25:03AM -0700, Christoph Lameter wrote:

> On Tue, 19 Sep 2006, Rohit Seth wrote:

>

> > For example, a user can run a batch job like backup inside containers.

> > This job if run unconstrained could step over most of the memory present

> > in system thus impacting other workloads running on the system at that

> > time. But when the same job is run inside containers then the backup

> > job is run within container limits.

>

> I just saw this for the first time since linux-mm was not cced. We have

> discussed a similar mechanism on linux-mm.

>

> We already have such a functionality in the kernel its called a cpuset. A

> container could be created simply by creating a fake node that then

> allows constraining applications to this node. We already track the

> types of pages per node. The statistics you want are already existing.

> See /proc/zoneinfo and /sys/devices/system/node/node*/*.

>

> > We use the term container to indicate a structure against which we track

> > and charge utilization of system resources like memory, tasks etc for a

> > workload. Containers will allow system admins to customize the

> > underlying platform for different applications based on their

> > performance and HW resource utilization needs. Containers contain

> > enough infrastructure to allow optimal resource utilization without

> > bogging down rest of the kernel. A system admin should be able to

> > create, manage and free containers easily.

>

> Right thats what cpusets do and it has been working fine for years. Maybe

> Paul can help you if you find anything missing in the existing means to

> control resources.

What I like about Rohit's patches is the page tracking stuff which seems quite simple but capable.

I suspect cpusets don't quite provide enough features for non-exclusive use of memory (eg. page tracking for directed reclaim).
