

---

Subject: Re: [PATCH 1/4] Virtualization/containers: introduction

Posted by [dev](#) on Tue, 07 Feb 2006 12:12:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

>>The important properties of the proposed container implementation:

>>- each container has unique ID in the system

> What namespace does this ID live in?

global namespace. can be virtualized later.

can be optional.

But the idea is simple. Eventually you will need some management tools anyway. And they should be able to refer to containers.

>>- each process in the kernel can belong to one container only

> Reasonable.

>

>

>>- effective container pointer (econtainer()) is used on the task to avoid

>>insertion of additional argument "container" to all functions where it is

>>required.

> Why is that desirable?

It was discussed with Linus and the reason is provided in this text actually.

There are 2 ways:

- to add additional argument "container" to all the functions where it is required.

Drawbacks: a) lot's of changes, b) compilation without virtualization is not the same. c) increased stack usage

- to add effective container pointer on the task. i.e. context which kernel should be in when works with virtualized resources.

Drawbacks: a) there are some places where you need to change effective container context explicitly such as TCP/IP.

>>- kernel compilation with disabled virtualization should result in old good

>>linux kernel

>

> A reasonable goal.

>

> Why do we need a container structure to hold pointers to other pointers? can't catch what you mean :) is it prohibited somehow? :)))

> May I please be added to the CC list.

> We are never going to form a consensus if all of the people doing implementations don't talk.

To make a consensus people need to make mutual concessions... Otherwise these talks are useless.

Kirill

---