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

Posted by [ebiederm](#) on Wed, 08 Feb 2006 17:16:45 GMT

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

Kirill Korotaev <dev@sw.ru> writes:

> Additional negative sides:

> - full isolation can be inconvenient from containers management point of
> view. You will need to introduce new modified tools such as top/ps/kill and many
> many others. You won't be able to strace/gdb processes from the host also.

Ok. I have to pick a nit here. Needing all new tools top/ps/kill was an artifact of your implementation. Mine does not suffer from it.

> - overhead when virtualization is off, result is not the same.
> - additional args everywhere (stack usage, etc.)

Agreed. When using a PID namespace the code always behaves the same. Which is with more arguments than the code used to have. However the code always behaving the same is a tremendous advantage for maintainability.

>> The vpid approach has the drawbacks of having to identify the conversion spots
>> of all vpid vs. pid semantics. On the otherhand it does take advantage
>> of the fact that no virtualization has to take place until a "container"
>> has been migrated, thus rendering most of the vpid<->pid calls to be
>> noops.

> It has some other additional advantages:

> - flexible: you can select full isolation or weak is required. I really believe
> this is very important.

I am willing to be convinced but so far there seem to be other solutions like running gdb_stubs inside your container. To address most of the issues.

In a fairly real sense mucking with a container from the outside appears to be a sysadmin layering violation.

>> The container is just an umbrella object that ties every "virtualized"
>> subsystem
>> together.

> Yep. And containers were what I wanted to start with actually. Not VPIDs.

The historical linux approach is to build things out of tasks sharing resources that give the impression of containers not out containers and their children. I am still trying to understand why that model does not work in this instance.

Eric
