Subject: Re: [RFC] Container mini-summit agenda for Sept 3, 2007
Posted by Oren Laadan on Fri, 31 Aug 2007 18:20:50 GMT
View Forum Message <> Reply to Message

Cedric Le Goater wrote:
> Hello Oren,
>
> Oren Laadan wrote:
>> Cedric Le Goater wrote:
>>> Hello All,
>>>
>>> Some of us will meet next week for the first mini-summit on containers.
>>> Many thanks to Alasdair Kergon and LCE for the help they provided in
>>> making this mini-summit happen !
>>>
>>> It will be help on Monday the 3rd of September from 9:00 to 12:45 at LCE
>>> in room D. We also might get a phone line for external participants and,
>>> if not, we should be able to set up a skype phone.
>>>
>>> Here's a first try for the Agenda.
>>>
>>> Global items
>>>
>>> [let's try to defer discussion after presentation]
>>>
>>> * Pavel Emelianov status update
>>> * Serge E. Hallyn Container Roadmap including
>>> . task containers (Paul Menage)
>>> . resource management (Srivatsa Vaddagiri)
>>>
>>> Special items
>>>
>>> [brainstorm sessions which we would like to focus on]
>>>
>>> * builing the global container object ('a la' openvz or vserver)
>>> * container user space tools
>>> * container checkpoint/restart
>>   5. checkpoint/restart
>>   memory c/r
>>   (there are a few designs and prototypes)
>>   (though this may be ironed out by then)
>>   per-container swapfile?
>>   overall checkpoint strategy (one of:)
>>   in-kernel
>>   userspace-driven
>>   hybrid
>>   overall restart strategy
>>   use freezer API

>> use suspend-to-disk?
>>
>> sysvipc
>> "set identifier" syscall
>> pid namespace
>> clone_with_pid()
>> There are other identifiers - pseudo terminals, message queues (mq)
>
> right, we have plans for developing these if needed (cf 2.)
>
>> (if you insist on supporting these ...). In general, we need a way
>> to specify the virtual id of a resource that is created.
>
> right, pierre peiffer has sent such a pachset for the sysvipc namespace.
> I'm looking at a clone_with_pid() for pid namespace.
>
>> I suggest
>> that this should be part of an interface between c/r and containers
>> (see below)
>>
>> live migration
>> aka pre-copy (which can be used for live migration but also to reduce
>> the downtime due to a checkpoint).
>
> yes that's usually what the buzz term "live migration" is used for.
>
>> how about adding incremental checkpoint to the list ?
>
> sure. I think it's a bit early to address these topic but we should have
> them in mind as some implementations already exist. And we need to gather
> all the needs.

exists in Zap; many lessons learned ;)

>
>> I think that it is also important to discuss an interface between c/r and
>> containers, each of which stands on it own. For instance, how to request
>> a specific virtual id (during restart), define required notifiers (to
>> set/unset c/r related data on/off a task), control c/r-related setting of
>> container (e.g. frozen, restarting) that may affect behavior, such as
>> signal handling, and so forth.
>
> This is exactly what we want to talk about.
>
> We need to identify these C/R needs, talk and agree about possible APIS
> and then convince the linux subsystem maintainers that they are useful
> for a large set of C/R solutions based on containers.
>

>> Also, such an interface can allow existing c/r implementations to work with
>> different virtualization implementations as they become available.
>
> what you call "virtualization" (private identifier namespaces), is I think
> being covered by the namespaces. These namespaces are not complete (like
> we're missing a way to reassign ids) but they are going in the right
> direction, IMO. However, I don't think there will be different
> "virtualization" implementations in mainline.

I do hope so too. I'm thinking that the current ones may take some time
to converge, and even then there may be out-of-mainline (experimental ?
alternative ?) implementation as it so happens with linux at time :)
In that case defining an interface can be useful (apart from the fact
that you tackle issues when you actually define one).
There is also the other side -- multiple c/r implementations (mainline
or not) that may be geared toward different goals depending on desires
performance, functionality etc.

>
>> Many of these were discussed in a recent Zap paper present in USENIX:
>> http://www.ncl.cs.columbia.edu/publications/usenix2007_fordist.pdf
>> The paper describes important design choices in Zap (but I'm biased ...).
>> I think it may serve as an appetizer for the discussion :P
>
> Thanks, I hope we all have time to read it.
>
> C.