Subject: containers development plans (July 10 version)
Posted by serge on Tue, 10 Jul 2007 21:39:43 GMT
View Forum Message <> Reply to Message

(If you missed earlier parts of this thread, you can catch earlier parts of this thread starting at
https://lists.linux-foundation.org/pipermail/containers/2007 -July/005860.html)

Thanks for all the recent feedback. I particularly added a lot from Paul Menage and Cedric.

We are trying to create a roadmap for the next year of 'container' development, to be reported to the upcoming kernel summit. Containers here is a bit of an ambiguous term, so we are taking it to mean all of:

1. namespaces
   kernel resource namespaces to support resource isolation and virtualization for virtual servers and application checkpoint/restart.
2. task containers framework
   the task containers (or, as Paul Jackson suggests, resource containers) framework by Paul Menage which especially provides a framework for subsystems which perform resource accounting and limits.
3. checkpoint/restart

A (still under construction) list of features we expect to be worked on next year looks like this:

1. completion of ongoing namespaces
   - pid namespace
     - merge two patchsets
     - clone_with_pid()
     - kthread cleanup
       - especially nfs
       - autofs
     - af_unix credentials (stores pid_t?)
   - net namespace
   - ro bind mounts
   - sysvipc
     - "set identifier" syscall
2. continuation with new namespaces
   - devpts, console, and ttydrivers
   - user
   - time
   - namespace management tools
   - namespace entering (using one of:)

bind_ns()
ns container subsystem
(vs refuse this functionality)
multiple /sys mounts
break /sys into smaller chunks?
shadow dirs vs namespaces
multiple proc mounts
likely need to extend on the work done for pid namespaces
i.e. other /proc files will need some care
3. any additional work needed for virtual servers?
i.e. in-kernel keyring usage for cross-usernamespace permissions, etc
nfs and rpc updates needed?
general security fixes
4. task containers functionality
base features
virtualized continerfs mounts
to support vserver mgmnt of sub-containers
locking cleanup
control file API simplification
control file prefixing with subsystem name
specific containers
usespace RBCE to provide controls for
users
groups
pgrp
executable
split cpusets into
cpuset
memset
network
connect/bind/accept controller using iptables
network flow id control
userspace per-container OOM handler

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?

In the list of stakeholders, I try to guess based on past comments and

contributions what *general* area they are most likely to contribute in.
I may try to narrow those down later, but am just trying to get something
out the door right now before my next computer breaks.

Stakeholders:
- Eric Biederman
  - everything
- google
  - containers
- ibm
  - everything
- kerlabs
  - checkpoint/restart
- openvz
  - everything
- osdl (Masahiko Takahashi?)
  - checkpoint/restart
- Linux-VServer
  - namespaces+containers
- zap project
  - checkpoint/restart
- planetlab
  - everything
- hp
  - ?
- XtreemOS
  - checkpoint/restart

Is anyone else still missing from the list?

thanks,
-serge

Subject: Re: containers development plans (July 10 version)
Posted by KAMEZAWA Hiroyuki on Tue, 10 Jul 2007 22:49:03 GMT
View Forum Message <> Reply to Message

On Tue, 10 Jul 2007 16:39:43 -0500
"Serge E. Hallyn" <serge@hallyn.com> wrote:
> In the list of stakeholders, I try to guess based on past comments and
> contributions what *general* area they are most likely to contribute in.
> I may try to narrow those down later, but am just trying to get something
> out the door right now before my next computer breaks.
>
> Stakeholders:
> Eric Biederman
> everything

> google
> containers
> ibm
> everything
> kerlabs
> checkpoint/restart
> openvz
> everything
> osdl (Masahiko Takahashi?)
> checkpoint/restart
> Linux-VServer
> namespaces+containers
> zap project
> checkpoint/restart
> planetlab
> everything
> hp
> ?
> XtreemOS
> checkpoint/restart
>
> Is anyone else still missing from the list?
>
hello,
I'm sorry if I misunderstand meaning of Stakeholders.
Recently, we (fujitsu+VA Linux Japan) made CKRM theam disperse and starts a new team
for Containers. We are mainly interested in resource control (cpu and memory).
Honestly, I'm now just studying patches for this area.
I'm glad if our team will be able to make contribution to this project in future.

Thanks,
 Hiroyuki Kamezawa.

Subject: Re: containers development plans (July 10 version)
Posted by Andrew Morton on Tue, 10 Jul 2007 23:49:49 GMT
View Forum Message <> Reply to Message

On Tue, 10 Jul 2007 16:39:43 -0500
"Serge E. Hallyn" <serge@hallyn.com> wrote:

> We are trying to create a roadmap for the next year of
> 'container' development, to be reported to the upcoming kernel
> summit. Containers here is a bit of an ambiguous term, so we are

> taking it to mean all of:
>
> 1. namespaces
> kernel resource namespaces to support resource isolation
> and virtualization for virtual servers and application
> checkpoint/restart.
> 2. task containers framework
> the task containers (or, as Paul Jackson suggests, resource
> containers) framework by Paul Menage which especially
> provides a framework for subsystems which perform resource
> accounting and limits.
> 3. checkpoint/restart

I would suggest that this material be fleshed out quite a lot with usage scenarios, applications, etc. Something which will help the kernel team at large understand the *value* of this work: what it offers our users.

Because people generally don't know that stuff yet. And if one starts explaining all this complexity, bug-potentiality and overhead-potentiality to kernel developers without first making it very clear what it will be gaining us, they will revolt.

IOW: do not presume that people want *any* of this stuff at this stage. First up, they need to get all fired up about how cool it all will be ;)

Subject: Re: containers development plans (July 10 version)
Posted by Paul Menage on Wed, 11 Jul 2007 06:06:42 GMT
View Forum Message <> Reply to Message

On 7/10/07, Serge E. Hallyn <serge@hallyn.com> wrote:
>
> A (still under construction) list of features we expect to be worked on
> next year looks like this:
> 4. task containers functionality
> specific containers

A couple of more container subsystem requests that have come out of the Linux Foundation Japan symposium, although I think they've also been mentioned before more than once - per-container swap and disk I/O scheduling.

I'm not familiar enough with the current Linux disk scheduler code to know how easy/hard it is to add rate guarantees on a per-container basis, but the swap one should be easier.

One potential issue with the swap container is how integrated should it be with the memory controller? I can certainly see people wanting

to be able to use a swap controller without requiring a page-based
memory controller (e.g. you might want to combine it with node-based
control via cpusets instead) but adding two pointers to the mm_struct,
one for swap controller subsystem and one for memory controller
subsystem, seems a little bit ugly.

Paul

Subject: Re: containers development plans (July 10 version)
Posted by Balbir Singh on Wed, 11 Jul 2007 06:31:57 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 7/10/07, Serge E. Hallyn <serge@hallyn.com> wrote:
>>
>> A (still under construction) list of features we expect to be worked on
>> next year looks like this:
>> 4. task containers functionality
>> specific containers
>
> A couple of more container subsystem requests that have come out of
> the Linux Foundation Japan symposium, although I think they've also
> been mentioned before more than once - per-container swap and disk I/O
> scheduling.
>

I think per container swap is interesting

> I'm not familiar enough with the current Linux disk scheduler code to
> know how easy/hard it is to add rate guarantees on a per-container
> basis, but the swap one should be easier.
>
> One potential issue with the swap container is how integrated should
> it be with the memory controller? I can certainly see people wanting
> to be able to use a swap controller without requiring a page-based
> memory controller (e.g. you might want to combine it with node-based
> control via cpusets instead) but adding two pointers to the mm_struct,
> one for swap controller subsystem and one for memory controller
> subsystem, seems a little bit ugly.
>

Well, it depends on how you define ugly. We could so something like
the namespace approach, have something like

struct mem_container_ptrs {
 swap_list;
 mem_container_ptr;

};

Although, I agree that per container swap is important, I feel that
we should add in the functionality, once we have basic page based
memory controller. It would make the whole setup easier to test
for functionality and performance.

> Paul

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

Subject: Re: containers development plans (July 10 version)
Posted by Paul Menage on Wed, 11 Jul 2007 06:55:03 GMT
View Forum Message <> Reply to Message

On 7/10/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>
> Well, it depends on how you define ugly. We could so something like
> the namespace approach, have something like
>
> struct mem_container_ptrs {
> swap_list;
> mem_container_ptr;
> };

I'm not quite sure what you're aiming for there. What would swap_list
represent?

I'm wondering if for both the per-page controller and the swap
controller, it would make sense to have a pointer back to an
appropriate process so we could get at a container pointer

Maybe something like:

- when an mm is created, store a pointer to the task_struct that it belongs to
- when a process exits and its mm_struct points to it, and there are
other mm users (i.e. a thread group leader exits before some of its
children), then find a different process that's using the same mm
(which will almost always be the next process in the list running
through current->tasks, but in strange situations we might need to
scan the global tasklist)

Then rather than having to have a pointer in the mm for either the page controller or the swap controller (and the consequent hassles of having refcounts from mm_structs to containers), you can just use the container membership of mm->owner.

>
> Although, I agree that per container swap is important, I feel that
> we should add in the functionality, once we have basic page based
> memory controller. It would make the whole setup easier to test
> for functionality and performance.

We don't really need to wait for a working page-based memory controller to be able to test a swap controller - cpusets gives memory controls too, albeit on a much coarser granularity.

Paul

Subject: Re: containers development plans (July 10 version)
Posted by Balbir Singh on Wed, 11 Jul 2007 07:21:17 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 7/10/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>>
>> Well, it depends on how you define ugly. We could so something like
>> the namespace approach, have something like
>>
>> struct mem_container_ptrs {
>>   swap_list;
>>   mem_container_ptr;
>> };
>
> I'm not quite sure what you're aiming for there. What would swap_list
> represent?
>

swap_list is a list of swap_devices associated with the container.
the mem_container_ptr points to the mem_container which inturn knows which container it belongs to.

> I'm wondering if for both the per-page controller and the swap
> controller, it would make sense to have a pointer back to an
> appropriate process so we could get at a container pointer
>
> Maybe something like:
>
> - when an mm is created, store a pointer to the task_struct that it

> belongs to
> - when a process exits and its mm_struct points to it, and there are
> other mm users (i.e. a thread group leader exits before some of its
> children), then find a different process that's using the same mm
> (which will almost always be the next process in the list running
> through current->tasks, but in strange situations we might need to
> scan the global tasklist)
>

We'll that sounds like a complicated scheme.

> Then rather than having to have a pointer in the mm for either the
> page controller or the swap controller (and the consequent hassles of
> having refcounts from mm_structs to containers), you can just use the
> container membership of mm->owner.
>

We do that currently, our mm->owner is called mm->mem_container. It points
to a data structure that contains information about the container to which
the mm belongs. The problem I see with mm->owner is that several threads
can belong to different containers. I see that we probably mean the same
thing, except that you suggest using a pointer to the task_struct from
mm_struct, which I am against in principle, due to the complexity of
changing owners frequently if the number of threads keep exiting at
a rapid rate.

>>
>> Although, I agree that per container swap is important, I feel that
>> we should add in the functionality, once we have basic page based
>> memory controller. It would make the whole setup easier to test
>> for functionality and performance.
>
> We don't really need to wait for a working page-based memory
> controller to be able to test a swap controller - cpusets gives memory
> controls too, albeit on a much coarser granularity.
>

We have a working page-based memory controller, it is yet to find it's way
into -mm though. The implementation of per container swap would be useful
and should work for both nevertheless. For cpusets to provide memory
control on non NUMA machines, we need to get in fuke numa emulation
support into all architectures.

> Paul

--

Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

Subject: Re: containers development plans (July 10 version)
Posted by Masahiko Takahashi on Wed, 11 Jul 2007 11:50:19 GMT
View Forum Message <> Reply to Message

On Tue, 2007-07-10 at 16:39 -0500, Serge E. Hallyn wrote:
> Stakeholders:
> Eric Biederman
> everything
> google
> containers
> ibm
> everything
> kerlabs
> checkpoint/restart
> openvz
> everything
> osdl (Masahiko Takahashi?)
> checkpoint/restart
> Linux-VServer
> namespaces+containers
> zap project
> checkpoint/restart
> planetlab
> everything
> hp
> ?
> XtreemOS
> checkpoint/restart

Serge,

Please change "osdl (Masahiko?)" to "NEC" because:
 o I'm a visiting engineer in OSDL (The Linux Foundation, now)
  and my activity on container is not OSDL's official one.
 o My visiting will end in this month but I will keep working
  on C/R from my company, NEC Japan.

Unfortunately(?) I don't belong to XtreemOS that Erich Focht
from NEC Europe does.

Thanks,

Masahiko.

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: containers development plans (July 10 version)
Posted by serue on Wed, 11 Jul 2007 13:32:10 GMT
View Forum Message <> Reply to Message

Quoting KAMEZAWA Hiroyuki (kamezawa.hiroyu@jp.fujitsu.com):
> On Tue, 10 Jul 2007 16:39:43 -0500
> "Serge E. Hallyn" <serge@hallyn.com> wrote:
> > In the list of stakeholders, I try to guess based on past comments and
> > contributions what *general* area they are most likely to contribute in.
> > I may try to narrow those down later, but am just trying to get something
> > out the door right now before my next computer breaks.
> >
> > Stakeholders:
> > Eric Biederman
> > everything
> > google
> > containers
> > ibm
> > everything
> > kerlabs
> > checkpoint/restart
> > openvz
> > everything
> > osdl (Masahiko Takahashi?)
> > checkpoint/restart
> > Linux-VServer
> > namespaces+containers
> > zap project
> > checkpoint/restart
> > planetlab
> > everything
> > hp
> > ?
> > XtreemOS
> > checkpoint/restart
> >
> > Is anyone else still missing from the list?
> >
> hello,

> I'm sorry if I misunderstand meaning of Stakeholders.

Yeah it's a bit ambiguous - it's basically people interested enough that
they may end up contributing something, or who already have existing
code they could help port and push to mainline.

> Recently, we (fujitsu+VA Linux Japan) made CKRM theam disperse and starts a new team
> for Containers. We are mainly interested in resource control (cpu and memory).
> Honestly, I'm now just studying patches for this area.
> I'm glad if our team will be able to make contribution to this project in future.

Great, I'll add you to the list :)

thanks,
-serge

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: containers development plans (July 10 version)
Posted by serue on Wed, 11 Jul 2007 13:35:36 GMT
View Forum Message <> Reply to Message

Quoting Balbir Singh (balbir@linux.vnet.ibm.com):
> Paul Menage wrote:
> > On 7/10/07, Serge E. Hallyn <serge@hallyn.com> wrote:
> >>
> >> A (still under construction) list of features we expect to be worked on
> >> next year looks like this:
> >> 4. task containers functionality
> >> specific containers
> >
> > A couple of more container subsystem requests that have come out of
> > the Linux Foundation Japan symposium, although I think they've also
> > been mentioned before more than once - per-container swap and disk I/O
> > scheduling.
> >
>
> I think per container swap is interesting

If we go with Dave Hansen's memory checkpoint technique, then per
container swap will be necessary anyway. So I had it listed under
"5. checkpoint/restart". I guess I could copy it, move it, or just
pull it into it's own item altogether.

thanks,

-serge

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: containers development plans (July 10 version)
Posted by serue on Wed, 11 Jul 2007 13:41:21 GMT
View Forum Message <> Reply to Message

Quoting Andrew Morton (akpm@linux-foundation.org):
> On Tue, 10 Jul 2007 16:39:43 -0500
> "Serge E. Hallyn" <serge@hallyn.com> wrote:
>
> > We are trying to create a roadmap for the next year of
> > 'container' development, to be reported to the upcoming kernel
> > summit. Containers here is a bit of an ambiguous term, so we are
> > taking it to mean all of:
> >
> > 1. namespaces
> > kernel resource namespaces to support resource isolation
> > and virtualization for virtual servers and application
> > checkpoint/restart.
> > 2. task containers framework
> > the task containers (or, as Paul Jackson suggests, resource
> > containers) framework by Paul Menage which especially
> > provides a framework for subsystems which perform resource
> > accounting and limits.
> > 3. checkpoint/restart
>
> I would suggest that this material be fleshed out quite a lot with usage
> scenarios, applications, etc. Something which will help the kernel team at
> large understand the *value* of this work: what it offers our users.
>
> Because people generally don't know that stuff yet. And if one starts
> explaining all this complexity, bug-potentiality and overhead-potentiality
> to kernel developers without first making it very clear what it will be
> gaining us, they will revolt.
>
> IOW: do not presume that people want *any* of this stuff at this stage.
> First up, they need to get all fired up about how cool it all will be ;)

Ah yes, great point, thanks for the suggestion.

If people who are have been working with each of these areas could send
me their use cases, that would be immensely helpful. Especially those
people who have been providing their own patches or product and know how

their customers were using them. I'm certain Cedric and probably Eric
can provide some good justification/motivation for checkpoint restart,
for example, and Kirill and Herbert for virtual servers.

thanks,
-serge

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: containers development plans (July 10 version)
Posted by serue on Wed, 11 Jul 2007 14:27:48 GMT
View Forum Message <> Reply to Message

Quoting Masahiko Takahashi (masahiko@linux-foundation.org):
> On Tue, 2007-07-10 at 16:39 -0500, Serge E. Hallyn wrote:
> > Stakeholders:
> > Eric Biederman
> > everything
> > google
> > containers
> > ibm
> > everything
> > kerlabs
> > checkpoint/restart
> > openvz
> > everything
> > osdl (Masahiko Takahashi?)
> > checkpoint/restart
> > Linux-VServer
> > namespaces+containers
> > zap project
> > checkpoint/restart
> > planetlab
> > everything
> > hp
> > ?
> > XtreemOS
> > checkpoint/restart
>
> Serge,
>
> Please change "osdl (Masahiko?)" to "NEC" because:
> o I'm a visiting engineer in OSDL (The Linux Foundation, now)
> and my activity on container is not OSDL's official one.
> o My visiting will end in this month but I will keep working

> on C/R from my company, NEC Japan.

Ok, thanks, will change that.

-serge

> Unfortunately(?) I don't belong to XtreemOS that Erich Focht
> from NEC Europe does.
>
>
> Thanks,
>
> Masahiko.
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: containers development plans (July 10 version)
Posted by dev on Thu, 12 Jul 2007 13:51:35 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> (If you missed earlier parts of this thread, you can catch earlier parts of
> this thread starting at
> https://lists.linux-foundation.org/pipermail/containers/2007 -July/005860.html)
>
> Thanks for all the recent feedback. I particularly added a lot from Paul
> Menage and Cedric.
>
> We are trying to create a roadmap for the next year of
> 'container' development, to be reported to the upcoming kernel
> summit. Containers here is a bit of an ambiguous term, so we are
> taking it to mean all of:
>
> 1. namespaces
> kernel resource namespaces to support resource isolation
> and virtualization for virtual servers and application
> checkpoint/restart.
> 2. task containers framework
> the task containers (or, as Paul Jackson suggests, resource
> containers) framework by Paul Menage which especially

> provides a framework for subsystems which perform resource
> accounting and limits.
> 3. checkpoint/restart
>
> A (still under construction) list of features we expect to be worked on
> next year looks like this:
>
> 1. completion of ongoing namespaces
> pid namespace
> merge two patchsets
sukadev@ and Pavel already agreed and will resend it soon
> clone_with_pid()
> kthread cleanup
> especially nfs
> autofs
> af_unix credentials (stores pid_t?)
> net namespace
> ro bind mounts

IMHO ro bind mounts are not related to namespaces anyhow, but ok if you guys want to mention it.

> sysvipc
> "set identifier" syscall

the last one is related to checkpointing, so plz move it from here...

> 2. continuation with new namespaces
> devpts, console, and ttydrivers
> user
> time
> namespace management tools
> namespace entering (using one of:)
> bind_ns()
> ns container subsystem
> (vs refuse this functionality)
> multiple /sys mounts
> break /sys into smaller chunks?
> shadow dirs vs namespaces
> multiple proc mounts
> likely need to extend on the work done for pid namespaces
> i.e. other /proc files will need some care

different statistics virtualization here in /proc for top and other tools

> 3. any additional work needed for virtual servers?
> i.e. in-kernel keyring usage for cross-usernamespace permissions, etc
> nfs and rpc updates needed?

> general security fixes

what is meant by "general security fixes"?

what I see additionaly:
 - device access controls (e.g. root in container should not have access to /dev/sda by default)
 - filesystems access controls

> 4. task containers functionality
> base features
> virtualized continerfs mounts
> to support vserver mgmnt of sub-containers
> locking cleanup
> control file API simplification
> control file prefixing with subsystem name
> specific containers
> usespace RBCE to provide controls for
> users
> groups
> pgrp
> executable
> split cpusets into
> cpuset
> memset
> network
> connect/bind/accept controller using iptables
> network flow id control
> userspace per-container OOM handler

I don't see much about resource management here at all.
We need resource controls for a lot of stuff like
- RSS
- kernel memory and different parameters like number of tasks
- disk quota
- disk I/O
- CPU fairness
- CPU limiting
- container aware OOM

imho it is all related and should be discussed.

> 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?
>
> In the list of stakeholders, I try to guess based on past comments and
> contributions what *general* area they are most likely to contribute in.
> I may try to narrow those down later, but am just trying to get something
> out the door right now before my next computer breaks.
>
> Stakeholders:
> Eric Biederman
> everything
> google
> containers
> ibm
> everything
> kerlabs
> checkpoint/restart
> openvz
> everything
> osdl (Masahiko Takahashi?)
> checkpoint/restart
> Linux-VServer
> namespaces+containers
> zap project
> checkpoint/restart
> planetlab
> everything
> hp
> ?
> XtreemOS
> checkpoint/restart
>
> Is anyone else still missing from the list?
>
> thanks,
> -serge
>

Subject: Re: containers development plans (July 10 version)
Posted by Srivatsa Vaddagiri on Thu, 12 Jul 2007 14:45:57 GMT
View Forum Message <> Reply to Message

On Thu, Jul 12, 2007 at 05:51:35PM +0400, Kirill Korotaev wrote:
> I don't see much about resource management here at all.

> We need resource controls for a lot of stuff like
> - RSS
> - kernel memory and different parameters like number of tasks
> - disk quota
> - disk I/O
> - CPU fairness
> - CPU limiting
> - container aware OOM
>
> imho it is all related and should be discussed.

Kirill,
 Definitely. I was supposed to provide a roadmap to Serge on our
resource management plans, which I haven't gotten around to provide yet.
Mostly, our interest is with cpu, memory and disk I/O controls (which
you have listed above). How abt putting together a roadmap on resource
management that can be used for kernel-summit discussion? I will send
out later a tentative roadmap for resource management that we are planning on.

--
Regards,
vatsa

Subject: Re: containers development plans (July 10 version)
Posted by serge on Thu, 12 Jul 2007 18:40:14 GMT
View Forum Message <> Reply to Message

Quoting Srivatsa Vaddagiri (vatsa@linux.vnet.ibm.com):
> On Thu, Jul 12, 2007 at 05:51:35PM +0400, Kirill Korotaev wrote:
> > I don't see much about resource management here at all.
> > We need resource controls for a lot of stuff like
> > - RSS
> > - kernel memory and different parameters like number of tasks
> > - disk quota
> > - disk I/O
> > - CPU fairness
> > - CPU limiting
> > - container aware OOM
> >
> > imho it is all related and should be discussed.
>
> Kirill,
> Definitely. I was supposed to provide a roadmap to Serge on our
> resource management plans, which I haven't gotten around to provide yet.
> Mostly, our interest is with cpu, memory and disk I/O controls (which
> you have listed above). How abt putting together a roadmap on resource
> management that can be used for kernel-summit discussion? I will send

> out later a tentative roadmap for resource management that we are planning on.

That'd be great, thanks.

-serge

Subject: Re: containers development plans (July 10 version)
Posted by serge on Thu, 12 Jul 2007 18:45:00 GMT
View Forum Message <> Reply to Message

Quoting Kirill Korotaev (dev@sw.ru):
> Serge E. Hallyn wrote:
> > (If you missed earlier parts of this thread, you can catch earlier parts of
> > this thread starting at
> > https://lists.linux-foundation.org/pipermail/containers/2007 -July/005860.html)
> >
> > Thanks for all the recent feedback. I particularly added a lot from Paul
> > Menage and Cedric.
> >
> > We are trying to create a roadmap for the next year of
> > 'container' development, to be reported to the upcoming kernel
> > summit. Containers here is a bit of an ambiguous term, so we are
> > taking it to mean all of:
> >
> > 1. namespaces
> > kernel resource namespaces to support resource isolation
> > and virtualization for virtual servers and application
> > checkpoint/restart.
> > 2. task containers framework
> > the task containers (or, as Paul Jackson suggests, resource
> > containers) framework by Paul Menage which especially
> > provides a framework for subsystems which perform resource
> > accounting and limits.
> > 3. checkpoint/restart
> >
> > A (still under construction) list of features we expect to be worked on
> > next year looks like this:
> >
> > 1. completion of ongoing namespaces
> > pid namespace
> > merge two patchsets
> sukadev@ and Pavel already agreed and will resend it soon
> > clone_with_pid()
> > kthread cleanup
> > especially nfs
> > autofs
> > af_unix credentials (stores pid_t?)

> > net namespace
> > ro bind mounts
>
> IMHO ro bind mounts are not related to namespaces anyhow, but ok if you guys want to
mention it.

Hmm, yes it's more for the "userspace containers" - meaning the
userspace usage of namespaces. But I'm not sure it's worth breaking
that out.

> > sysvipc
> > "set identifier" syscall
>
> the last one is related to checkpointing, so plz move it from here...

It started under checkpointing, but I'll move it back :)

> > 2. continuation with new namespaces
> > devpts, console, and ttydrivers
> > user
> > time
> > namespace management tools
> > namespace entering (using one of:)
> > bind_ns()
> > ns container subsystem
> > (vs refuse this functionality)
> > multiple /sys mounts
> > break /sys into smaller chunks?
> > shadow dirs vs namespaces
> > multiple proc mounts
> > likely need to extend on the work done for pid namespaces
> > i.e. other /proc files will need some care
>
> different statistics virtualization here in /proc for top and other tools
>
> > 3. any additional work needed for virtual servers?
> > i.e. in-kernel keyring usage for cross-usernamespace permissions, etc
> > nfs and rpc updates needed?
> > general security fixes
>
> what is meant by "general security fixes"?

I think it means "we haven't thought it through enough" :)

For instance, something needs to be done to be able to hand
partial capabilities to admins in a container/virtual server. We've
talked about doing this using the in-kernel keyring, but we are far from
consensus or patches, and this will have to be solved.

> what I see additionaly:
> - device access controls (e.g. root in container should not have access to /dev/sda by default)

Yes, that kind of falls under the above, but I'll add it separately.

> - filesystems access controls

ditto.

> > 4. task containers functionality
> > base features
> > virtualized continerfs mounts
> > to support vserver mgmnt of sub-containers
> > locking cleanup
> > control file API simplification
> > control file prefixing with subsystem name
> > specific containers
> > usespace RBCE to provide controls for
> > users
> > groups
> > pgrp
> > executable
> > split cpusets into
> > cpuset
> > memset
> > network
> > connect/bind/accept controller using iptables
> > network flow id control
> > userspace per-container OOM handler
>
> I don't see much about resource management here at all.
> We need resource controls for a lot of stuff like
> - RSS
> - kernel memory and different parameters like number of tasks
> - disk quota
> - disk I/O
> - CPU fairness
> - CPU limiting
> - container aware OOM
>
> imho it is all related and should be discussed.
>
> > 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?
> >
> > In the list of stakeholders, I try to guess based on past comments and
> > contributions what *general* area they are most likely to contribute in.
> > I may try to narrow those down later, but am just trying to get something
> > out the door right now before my next computer breaks.
> >
> > Stakeholders:
> > Eric Biederman
> > everything
> > google
> > containers
> > ibm
> > everything
> > kerlabs
> > checkpoint/restart
> > openvz
> > everything
> > osdl (Masahiko Takahashi?)
> > checkpoint/restart
> > Linux-VServer
> > namespaces+containers
> > zap project
> > checkpoint/restart
> > planetlab
> > everything
> > hp
> > ?
> > XtreemOS
> > checkpoint/restart
> >
> > Is anyone else still missing from the list?
> >
> > thanks,
> > -serge
> >

thanks Kirill,

-serge

Subject: Re: containers development plans (July 10 version)
Posted by dev on Fri, 13 Jul 2007 13:10:46 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 7/10/07, Serge E. Hallyn <serge@hallyn.com> wrote:
>
>>A (still under construction) list of features we expect to be worked on
>>next year looks like this:
>> 4. task containers functionality
>> specific containers
>
>
> A couple of more container subsystem requests that have come out of
> the Linux Foundation Japan symposium, although I think they've also
> been mentioned before more than once - per-container swap and disk I/O
> scheduling.
>
> I'm not familiar enough with the current Linux disk scheduler code to
> know how easy/hard it is to add rate guarantees on a per-container
> basis, but the swap one should be easier.

Paul, OpenVZ implements 2-level disk I/O scheduler based on CFQ.
It is not that hard to implement and we will propose the patches
a bit later based on your containers stuff.

However, there is one big problem which I'm not sure mainstream
is ready to go with. Writes. Writes are usually asynchronous
and it is impossible to say which container caused the write
when it goes to the scheduler.
OpenVZ solves this by tracking dirty pages and context which made them dirty.
But it is a next step after std CFQ modifying.

> One potential issue with the swap container is how integrated should
> it be with the memory controller? I can certainly see people wanting
> to be able to use a swap controller without requiring a page-based
> memory controller (e.g. you might want to combine it with node-based
> control via cpusets instead) but adding two pointers to the mm_struct,
> one for swap controller subsystem and one for memory controller
> subsystem, seems a little bit ugly.

Thanks,
Kirill