Subject: Re: [Vserver] VServer vs OpenVZ
Posted by TheWiseOne on Sat, 10 Dec 2005 19:31:08 GMT
View Forum Message <> Reply to Message

Eugen Leitl wrote:
> Before I try OpenVZ I would like to hear comments of people
> who've ran both VServer and OpenVZ, preferrably on the same
> hardware, on how both compare.
>
> Factors of interest are stability, Debian support,
> hardware utilization, documentation and community support,
> security.
>

As the owner of one of the larger VPS providers around we do use Virtuozzo (commercial support, less work packaging, etc) although I am a big fan of the VServer project.

For a hosting service I'd have to recommend Virtuozzo over either, as is a large purpose of OpenVZ I believe :) If you feel you can do the work to keep track of kernel changes, participate actively in debugging, etc then VServer would be fine.

As far as the technical debate between the projects, I have more interest in inputting on that.

1) "Fair scheduling" - as far as I can tell the VZ "fair scheduler" does nothing the VServer QoS/Limit system does. If anything, the VZ fair scheduler is not yet O(1) which is a big negative. VServer is built on standard kernel and therefore uses the O(1) scheduler (an absolute must when you have so many processes running on a single kernel).

2) Networking - The VZ venet0 is not perfect (no IPv6, still limited iptables, etc), but it still allows a lot more to be performed than VServer in the networking arena.

3) Disk/memory sharing - OpenVZ has nothing. Virtuozzo uses an overlay fs "vzfs". The templates are good for an enterprise environment, but really prove useless in a hosting environment. vzfs is overlay and therefore suffers from double caching (it caches both files in /vz/private (backing) and /vz/root (mount)). VServer uses CoW links which are modified hard links and eliminates the double caching. The Vserver vunify program (to re-link identical files due to user upgrading to same RPM's across VPS's, etc) takes a few minutes to run. The Virtuozzo vzcache program to do the same can take 2+ days to run on a host with 60+ VPS's.

4) In most other areas OpenVZ and VServer are similar. OpenVZ has many

UBC's, but since most oversell some such as vmguarpages really have no affect. Vserver limits the major memory limits (with RSS being a key one that OpenVZ cannot do). On the flip side OpenVZ can limit lowmem (kmemsize) while VServer cannot.

5) Quota inside VPS - The new way of linking the quota user/group files to /proc in OpenVZ is very good (genius?) idea I think. Currently I'm not even 100% sure quota-inside-VPS works under VServer, it's been a few months since I've experimented with it or talked to Herbert.

So which is better? Neither, it depends what the user requires and what they are willing and wanting to do.

My 2 cents..

Thanks,
Matt Ayres

Subject: Re: Re: [Vserver] VServer vs OpenVZ
Posted by dev on Sun, 11 Dec 2005 09:41:14 GMT
View Forum Message <> Reply to Message

> Eugen Leitl wrote:
>
>> Before I try OpenVZ I would like to hear comments of people
>> who've ran both VServer and OpenVZ, preferrably on the same
>> hardware, on how both compare.
>> Factors of interest are stability, Debian support, hardware
>> utilization, documentation and community support,
>> security.
>>
>
> As the owner of one of the larger VPS providers around we do use
> Virtuozzo (commercial support, less work packaging, etc) although I am a
> big fan of the VServer project.
>
> For a hosting service I'd have to recommend Virtuozzo over either, as is
> a large purpose of OpenVZ I believe :) If you feel you can do the work
> to keep track of kernel changes, participate actively in debugging, etc
> then VServer would be fine.
I suppose OpenVZ can do the same :) if you like debugging :)

> As far as the technical debate between the projects, I have more
> interest in inputting on that.
>
> 1) "Fair scheduling" - as far as I can tell the VZ "fair scheduler" does
> nothing the VServer QoS/Limit system does. If anything, the VZ fair

> scheduler is not yet O(1) which is a big negative. VServer is built on
> standard kernel and therefore uses the O(1) scheduler (an absolute must
> when you have so many processes running on a single kernel).
this is not true! Fairscheduler in current implementation on 2.6 kernel
is O(1) and doesn't depend on number of processes anyhow!
And we are working on improving it much more and implement some
additional features in it.

> 2) Networking - The VZ venet0 is not perfect (no IPv6, still limited
> iptables, etc), but it still allows a lot more to be performed than
> VServer in the networking arena.
Mmm, if you are an SWsoft customer you can always request some netfilter
module to be virtualized. It's a matter of couple of days actually...
And there are no any problems with it. We just virtualized the basic set
of iptable matches and targets and those which are rarely used didn't.

There were only 4(!) IPv6 requests during 2 years. It's the same as
netfilters - no much demand, no feature.

> 3) Disk/memory sharing - OpenVZ has nothing. Virtuozzo uses an overlay
> fs "vzfs". The templates are good for an enterprise environment, but
> really prove useless in a hosting environment. vzfs is overlay and
> therefore suffers from double caching (it caches both files in
> /vz/private (backing) and /vz/root (mount)).
not sure what you mean... memory caching?! it is not true again then...

> VServer uses CoW links
> which are modified hard links and eliminates the double caching. The
> Vserver vunify program (to re-link identical files due to user upgrading
> to same RPM's across VPS's, etc) takes a few minutes to run. The
> Virtuozzo vzcache program to do the same can take 2+ days to run on a
> host with 60+ VPS's.

> 4) In most other areas OpenVZ and VServer are similar. OpenVZ has many
> UBC's, but since most oversell some such as vmguarpages really have no
> affect. Vserver limits the major memory limits (with RSS being a key
> one that OpenVZ cannot do). On the flip side OpenVZ can limit lowmem
> (kmemsize) while VServer cannot.
RSS is good yeah, but there are lot's of DoS possible if you limit RSS
only. No lowmem, no TCP bufs, etc... I personally know many ways of
DoSing of other resources, but if you don't care security this is
probably ok.

> 5) Quota inside VPS - The new way of linking the quota user/group files
> to /proc in OpenVZ is very good (genius?) idea I think. Currently I'm
> not even 100% sure quota-inside-VPS works under VServer, it's been a few
> months since I've experimented with it or talked to Herbert
Glad to hear that.

> So which is better? Neither, it depends what the user requires and what
> they are willing and wanting to do.
>
> My 2 cents..
thanks for a lot of information!

Kirill

Subject: Re: Re: [Vserver] VServer vs OpenVZ
Posted by TheWiseOne on Sun, 11 Dec 2005 20:03:16 GMT
View Forum Message <> Reply to Message

>>
>> 1) "Fair scheduling" - as far as I can tell the VZ "fair scheduler"
>> does nothing the VServer QoS/Limit system does. If anything, the VZ
>> fair scheduler is not yet O(1) which is a big negative. VServer is
>> built on standard kernel and therefore uses the O(1) scheduler (an
>> absolute must when you have so many processes running on a single
>> kernel).
> this is not true! Fairscheduler in current implementation on 2.6 kernel
> is O(1) and doesn't depend on number of processes anyhow!
> And we are working on improving it much more and implement some
> additional features in it.

Great, why not provide packages against latest Virtuozzo (with modules, vzfs, etc) for better real world testing? What numbers are you seeing in regards to load, based on my estimates with 3000 procs and an avg of 3 running on a server with a load average of 3 should drop to much lower. What kind of numbers do you see in your tests?

>
>> 3) Disk/memory sharing - OpenVZ has nothing. Virtuozzo uses an
>> overlay fs "vzfs". The templates are good for an enterprise
>> environment, but really prove useless in a hosting environment. vzfs
>> is overlay and therefore suffers from double caching (it caches both
>> files in /vz/private (backing) and /vz/root (mount)).
> not sure what you mean... memory caching?! it is not true again then...
>

The kernel caches based on inode number. If you modified the caching part of the module then I may be incorrect in my thinking. Take example:

ls -ai /vz/private/1/root/bin/ls
41361462 /vz/private/1/root/bin/ls
ls -ai /vz/template/redhat-as3-minimal/coreutils-4.5.3-26/bin/ls
1998864 /vz/template/redhat-as3-minimal/coreutils-4.5.3-26/bin/ls

ls -ai /vz/root/1/bin/ls
41361462 /vz/root/1/bin/ls

The kernel will cache both inodes 41361462 and 1998864. Knowing that, when I look at my host servers with 8GB of RAM and see 4GB being used for cache/buffers I get angry. vzfs appears to be a standard unionfs with support for CoW to those who do not see the source. You ignored responding to how VServer does it which results in using a patched kernel to have special CoW links without a union mount. The links are based on a hard link architecture resulting in 1 inode. Also commenting on was ignored vunify and vzcache speeds.

Pertaining paragraph:
>> VServer uses CoW links which are modified hard links and eliminates
>> the double caching. The Vserver vunify program (to re-link identical
>> files due to user upgrading to same RPM's across VPS's, etc) takes a
>> few minutes to run. The Virtuozzo vzcache program to do the same can
>> take 2+ days to run on a host with 60+ VPS's.
>
>
>
>> 4) In most other areas OpenVZ and VServer are similar. OpenVZ has
>> many UBC's, but since most oversell some such as vmguarpages really
>> have no affect. Vserver limits the major memory limits (with RSS
>> being a key one that OpenVZ cannot do). On the flip side OpenVZ can
>> limit lowmem (kmemsize) while VServer cannot.
> RSS is good yeah, but there are lot's of DoS possible if you limit RSS
> only. No lowmem, no TCP bufs, etc... I personally know many ways of
> DoSing of other resources, but if you don't care security this is
> probably ok.
>

It does RSS and VM limiting with no guarantees. It also does locked pages, sockets, etc. The argument of who has more structures to limit is actually rather pointless now as VServer could take the OpenVZ limits to see what they can limit and decide which they want to implement. That is only a matter of time. I'm sure Herbert has seen output of /proc/user_beancounters before OpenVZ was even released and didn't see a reason for some of the limits. What I was pointing out was differences currently. A very minor advantage of VServer if they virtualize the meminfo structure to reflect memory/swap total/usage based on the RSS/VM limits.

Thank you,
Matt Ayres

Subject: Re: Re: [Vserver] VServer vs OpenVZ
Posted by TheWiseOne on Sun, 11 Dec 2005 20:09:49 GMT
View Forum Message <> Reply to Message

> The kernel caches based on inode number. If you modified the caching
> part of the module then I may be incorrect in my thinking. Take example:
>
> # ls -ai /vz/private/1/root/bin/ls
> 41361462 /vz/private/1/root/bin/ls
> # ls -ai /vz/template/redhat-as3-minimal/coreutils-4.5.3-26/bin/ls
> 1998864 /vz/template/redhat-as3-minimal/coreutils-4.5.3-26/bin/ls
> # ls -ai /vz/root/1/bin/ls
> 41361462 /vz/root/1/bin/ls
>

This may actually be an error. This new info was on a linked file
(using templates). I tried on a VPS that has broken the link and found
this:

ls -ali /vz/private/101/root/bin/ls
245325878 -rwxr-xr-x 1 root root 67700 Sep 29 11:42
/vz/private/101/root/bin/ls
ls -ali /vz/root/101/bin/ls
245325878 -rwxr-xr-x 1 root root 67700 Sep 29 11:42
/vz/root/101/bin/ls

Or - the third thought is that we are better off not using the CoW/vzfs
part of Virtuozzo at all and OpenVZ is a more scalable alternative to
Virtuozzo.

Subject: Re: Re: [Vserver] VServer vs OpenVZ
Posted by dev on Mon, 12 Dec 2005 20:22:15 GMT
View Forum Message <> Reply to Message

>>> 1) "Fair scheduling" - as far as I can tell the VZ "fair scheduler"
>>> does nothing the VServer QoS/Limit system does. If anything, the VZ
>>> fair scheduler is not yet O(1) which is a big negative. VServer is
>>> built on standard kernel and therefore uses the O(1) scheduler (an
>>> absolute must when you have so many processes running on a single
>>> kernel)
>>
>> this is not true! Fairscheduler in current implementation on 2.6
>> kernel is O(1) and doesn't depend on number of processes anyhow!
>> And we are working on improving it much more and implement some
>> additional features in it.
>
> Great, why not provide packages against latest Virtuozzo (with modules,

> vzfs, etc) for better real world testing? What numbers are you seeing
> in regards to load, based on my estimates with 3000 procs and an avg of
> 3 running on a server with a load average of 3 should drop to much
> lower. What kind of numbers do you see in your tests?
1. Virtuozzo 3.0 with O(1) scheduler will be released very soon. OpenVZ already has it, so you can test it right now.
2. Can't understand your statement about 3000 procs/3 avg etc. What estimations do you mean? Can you describe it in more details?

>>> 3) Disk/memory sharing - OpenVZ has nothing. Virtuozzo uses an
>>> overlay fs "vzfs". The templates are good for an enterprise
>>> environment, but really prove useless in a hosting environment. vzfs
>>> is overlay and therefore suffers from double caching (it caches both
>>> files in /vz/private (backing) and /vz/root (mount)).
>>
>> not sure what you mean... memory caching?! it is not true again then...

> The kernel caches based on inode number. If you modified the caching
> part of the module then I may be incorrect in my thinking. Take example:
>
> # ls -ai /vz/private/1/root/bin/ls
> 41361462 /vz/private/1/root/bin/ls
> # ls -ai /vz/template/redhat-as3-minimal/coreutils-4.5.3-26/bin/ls
> 1998864 /vz/template/redhat-as3-minimal/coreutils-4.5.3-26/bin/ls
> # ls -ai /vz/root/1/bin/ls
> 41361462 /vz/root/1/bin/ls
1. Kernel doesn't cache anything based on inode numbers. Inode numbers are just a magic IDs. Nothing more. Internally everything is much more complex.
2. inode numbers can be different with vzfs, but the data cached under these inodes is the same, i.e. no double caching with vzfs happens. This is the main purpose of VZFS and this is required for scalability: to have only one instance of data in memory.

> The kernel will cache both inodes 41361462 and 1998864. Knowing that,
> when I look at my host servers with 8GB of RAM and see 4GB being used
> for cache/buffers I get angry.
Looks like you are misinterpreting /proc/meminfo output. Let me explain. /proc/meminfo shows you amount of memory used for caching of files and buffers, both of which are reclaimable. This means that:
- cached memory is not a wastage of you HW memory, it's a temporarely cache of disk files. The bigger the better.
- since it's reclaimable it's not a memory which is pinned downed and is freed on demand when some applications or kernel really need memory for its own use. i.e. it means that you have 4GB of _FREE_ RAM which kernel _temporarily_ used for caches. Why are you angry with it?! I would be happy in this situation :)
- as I wrote before the data is cached in memory only once, so in your

example with /bin/ls it takes only ~68k of data memory in caches +
dentry/inode caches (internal kernel structures). And raw figures in
/proc/meminfo doesn't allow to understand whether it was cached once or
twice or more times in memory. It's just 4GB of RAM which are used for
caches, no any other information here.
- on practice vzfs saves you 40-70% of VPS memory compared to OpenVZ
when VPSs are based on the same template. Sure, the more hungry VPS is
the less relative gain vzfs provide for such VPS.
So your comment about better scalability of OpenVZ than Virtuozzo looks
wrong to me...

> vzfs appears to be a standard unionfs
> with support for CoW to those who do not see the source. You ignored
> responding to how VServer does it which results in using a patched
> kernel to have special CoW links without a union mount. The links are
> based on a hard link architecture resulting in 1 inode. Also commenting
> on was ignored vunify and vzcache speeds.
It was not ignored actually, sorry that I didn't replied to it before
and made you think so.

We immediately started investigating your report. I believe vunify tool
is more like vzpkglink which is also fast enough. But it will be checked
more thoroughly.

More likely the whole vzcache will be reworked. I really appreciate such
reports and probably will return to you with more questions on it.

>> RSS is good yeah, but there are lot's of DoS possible if you limit RSS
>> only. No lowmem, no TCP bufs, etc... I personally know many ways of
>> DoSing of other resources, but if you don't care security this is
>> probably ok.
>>
> It does RSS and VM limiting with no guarantees. It also does locked
> pages, sockets, etc. The argument of who has more structures to limit
> is actually rather pointless now as VServer could take the OpenVZ limits
> to see what they can limit and decide which they want to implement. That
> is only a matter of time. I'm sure Herbert has seen output of
> /proc/user_beancounters before OpenVZ was even released and didn't see a
> reason for some of the limits. What I was pointing out was differences
> currently. A very minor advantage of VServer if they virtualize the
> meminfo structure to reflect memory/swap total/usage based on the RSS/VM
> limits.
meminfo will be fixed soon, I suppose.

Kirill