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