
Subject: Re: [Vserver] Re: Container Test Campaign

Posted by [kir](#) on Tue, 04 Jul 2006 16:30:26 GMT

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

See my comments below.

In general - please don't get the impression I try to be fastidious. I'm just trying to help you create a system in which results can be reproducible and trusted. There are a lot of factors that influence the performance; some of those are far from being obvious.

Clément Calmels wrote:

> A basic 'patch' test looks like:

> o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> example)

> o reboot

> o run dbench on /tmp with 8 processes

>

IMO you should add a reboot here, in between `_different_` tests, just because previous tests should not influence the following ones.

Certainly you do not need a reboot before iterations of the same test.

> o run tbench with 8 processes

> o run lmbench

> o run kernbench

>

> For test inside a 'guest' I just do something like:

> o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> example)

> o reboot

>

Here you do not have to reboot. OpenVZ tools does not require OpenVZ kernel to be built.

> o build the utilities (vzctl+vzquota for example)

> o reboot

> o launch a guest

>

Even this part is tricky! You haven't specified whether you create the guest before or after the reboot. Let me explain.

If you create a guest before the reboot, the performance (at least at the first iteration) could be a bit higher than if you create a guest after the reboot. The reason is in the second case the buffer cache will be filled with OS template data (which is several hundred megs).

> o run in the guest dbench ...

>

Again, a clean reboot is needed IMO.

> o run in the guest tbench ...

>

>
> -The results are the average value of several iterations of each set of
> these kind of tests.

Hope you do not recompile the kernels before the iterations (just to speed things up).

> I will try to update the site with the numbers of
> iterations behind each values.

>
Would be great to have that data (as well as the results of the individual iterations, and probably graphs for the individual iterations -- to see the "warming" progress, discrepancy between iterations, degradation over iterations (if that takes place) etc).

Based on that data, one can decide to further tailor the testing process. For example, if there are visible signs of "warming" for a first few iterations (i.e. the performance is worse) it makes sense to unconditionally exclude those from the results. If there is a sign of degradation, something is wrong. And so on...

> - For the filesystem testing, the partition is not reformatted. I can
> change this behaviour...

>
Disk layout is influencing the results of the test which do heavy I/O. Just a single example: if you try to test the performance of a web server, results will decrease over time. The reason of degradation is ... web server's access_log file! It grows over time, and write operation takes a bit longer (due to several different reasons).

The same will happen with most of the other tests involving I/O. Thus, test results will be non-accurate. To achieve more accuracy and exclude the impact of the disk and filesystem layout to the results, you should reformat the partition you use for testing each time before the test.

Note that you don't have to reinstall everything from scratch -- just use a separate partition (mounted to say /mnt/temptest) and make sure most of the I/O during the test happens on that partition.

> - For the settings of the guest I tried to use the default settings (I
> had to change some openvz guest settings) just following the HOWTO on
> vserver or openvz site.

> For the kernel parameters, did you mean kernel config file tweaking?

>
No I mean those params from /proc/sys (== /etc/sysctl.conf). For example, if you want networking for OpenVZ guests, you have to turn on ip_forwarding. There are some params affecting network performance, such as various gc_thresholds. For the big number of guests, you have to tune some system-wide parameters as well.

So I am leading to the proposition that all such changes should be documented in test results.

> - Cron are stopped during tests.

>

Hope you do that for the guest as well... :)

> - All binaries are always build in the test node.

>

I assuming you are doing your tests on the same system (i.e. same compiler/libs/whatever else), and you do not change that system over time (i.e. you do not upgrade gcc on it in between the tests).

> Feel free to provide me different scenario which you think are more

> relevant.
