Subject: "strong" Disk I/O prioritisation
Posted by gnutered on Wed, 09 Apr 2008 11:45:17 GMT
View Forum Message <> Reply to Message

I want to be able to give absolute I/O priority to one (or a few) VE, and have it so that all I/O for other VEs gets serviced only when idle.

I tried using the "set ... --ioprio" option, as per below, but found that in rough tests, it only slews things slightly in favour of my preferred VE.

Am I doing something wrong? Is this as good as it gets?

As a workaround, I tried using ionice within the VE. Even after adding the sys_admin and sys_nice capabilities, I still got

ionice -c3 id
ioprio_set: Operation not permitted

I could perhaps live with running ionice within my low priority VEs if I could get it to work.

Detail...

Kernel: Linux sodium 2.6.18-028stab053.tl20080408 (my compile of OVZ's latest vanilla kernel + latest stable patch)

urand is a 512MB cat of /dev/urandom in the root directory of each of VEs 100 and 200 (different file)

By themselves, it takes about 8 seconds to cat the file:
tony@sodium:~$ sudo vzctl exec 100 time cat /urand \> /dev/null

real 0m7.754s
user 0m0.052s
sys 0m0.524s
tony@sodium:~$ sudo vzctl exec 200 time cat /urand \> /dev/null

real 0m7.803s
user 0m0.044s
sys 0m0.520s

I start running one in a constant loop:
tony@sodium:~$ while true; do sudo vzctl exec 100 time cat /urand \> /dev/null ; done

Then in another terminal I cat the file on the other VE. This shows that IO is roughly shared between VEs at the moment:
tony@sodium:~$ sudo vzctl exec 200 time cat /urand \> /dev/null

real 0m17.745s

user 0m0.068s
sys 0m0.504s

Now I set the priorities:
tony@sodium:~$ sudo vzctl set 100 --ioprio 0 --save
Saved parameters for VE 100
tony@sodium:~$ sudo vzctl set 200 --ioprio 7 --save
Saved parameters for VE 200

And restart the loop. In the other VE, I run once:
tony@sodium:~$ sudo vzctl exec 200 time cat /urand \> /dev/null

real 0m13.341s
user 0m0.080s
sys 0m0.488s

I was hoping this figure would be a lot closer to the original 8 seconds. Can I do better than this in brutally prioritising one VE over another?

My rationale for this is to protect my production webserver (two heavily used phpbb websites) from my other VEs. Currently (linux-vserver, Ubuntu 6.06), simply copying a file of moderate size (more than 500MB) brings the prod webserver to its knees for minutes, which makes me unpopular.

Subject: Re: "strong" Disk I/O prioritisation
Posted by TheWiseOne on Wed, 09 Apr 2008 14:57:59 GMT
View Forum Message <> Reply to Message

I/O priorities in OpenVZ are proportional share so I don't think you can do what you want. There was a dm-band project released a little while ago that provides hard limits on disk I/O bandwidth for both Xen and OpenVZ.

http://kerneltrap.org/Linux/Dm-band_Block_IO_Bandwidth_Contr oller

Perhaps you could petition OpenVZ to include this, although from what I can tell from talking to Kirill they have a full plate for the next few months.

Subject: Re: "strong" Disk I/O prioritisation
Posted by Vasily Tarasov on Thu, 10 Apr 2008 06:34:08 GMT
View Forum Message <> Reply to Message

Hello Tony,

you wrote:
Quote:I want to be able to give absolute I/O priority to one (or a few) VE, and have it so that all I/O for other VEs gets serviced only when idle.

It is impossible to assign an absolute I/O priority to VE at the moment. When you use --ioprio option of vzctl, you assign a relative share of time, during which the VE in question can work with a block device.

Also current implementation doesn't support "idle" class of VEs, which are serviced only if nobody else uses the block device. This is in future plans.

you wrote
Quote:As a workaround, I tried using ionice within the VE. Even after adding the sys_admin and sys_nice capabilities, I still got
ionice -c3 id
ioprio_set: Operation not permitted

That's a good point! Thank you for noticing. The thing is that before prioritization was introduced in OpenVZ, sys_ioprio_set() system call (which is used by ionice utility) was prohibited inside VE for understandable reasons. But now, we can allow that!
You can comment
if (!ve_is_super(get_exec_env()))
return -EPERM; check in ./fs/ioprio.c file to check if it will help. I personally think, that setting a priority of the processes in VE will not help in your situation a lot.

Now several words about your tests and their results:
1) When you do cat for the first time, some parts or even the whole file is in cache. So, 2nd time you do cat, it is not read from the disk, but from the main memory. It can introduce significant distortions.
2) In implementation that works now in OpenVZ you can notice the effect of prioritization much more, if you will run not one "disk-reader" (as in your test) in VE, but several of them. It is not the feature, but a drawback, and we're working on improvements in this area.

you wrote:
Quote:Can I do better than this in brutally prioritising one VE over another? My rationale for this is to protect my production webserver (two heavily used phpbb websites) from my other VEs. Currently (linux-vserver, Ubuntu 6.06), simply copying a file of moderate size (more than 500MB) brings the prod webserver to its knees for minutes, which makes me unpopular.

I understand you rationale very good. The I/O prioritization in Linux is on the blooding edge, so there is not perfect solution now. If current OpenVZ prioritizations is not enough for you, you can move your production VE (or all other VEs) to a separate hard drive.

HTH!

Subject: Re: "strong" Disk I/O prioritisation
Posted by gnutered on Thu, 10 Apr 2008 10:32:30 GMT
View Forum Message <> Reply to Message

Vasily, thanks.

I'm recompiling the kernel now with that commenting out you mentioned.

Quote:1) When you do cat for the first time, some parts or even the whole file is in cache. So, 2nd time you do cat, it is not read from the disk, but from the main memory. It can introduce significant distortions.

Yes, I knew that - that's why the files are each 512MB (phys mem = 521MB too), and I never read the same one twice, and it's read sequentially. When I run the test I seem to get consistent numbers.

Quote:2) In implementation that works now in OpenVZ you can notice the effect of prioritization much more, if you will run not one "disk-reader" (as in your test) in VE, but several of them. It is not the feature, but a drawback, and we're working on improvements in this area.

I think you're saying that my tests aren't really real-world representations, and the performance might be better on my production server, and I agree. I'm going to install a copy of the production webserver on this box, and try it then.

I look forward to future improvements in the I/O prioritisation.

I'll also get to play with the different schedulers that way too.

Tony