Subject: *SOLVED* IO scheduling
Posted by dagr on Wed, 20 Sep 2006 18:25:32 GMT
View Forum Message <> Reply to Message

It looks like vserver has io scheduler (I/O scheduler queue per context)
http://linux-vserver.org/ChangeLog-2.1
Does openvz or virtuozzo have smth like this ? I havent found it in UBC list, although it should be as you try to prevent dos attacks.

Subject: Re: IO scheduling
Posted by Vasily Tarasov on Thu, 21 Sep 2006 06:43:40 GMT
View Forum Message <> Reply to Message

Hi,

I'll try to explain you what is in Vserver concerning IO scheduling and why it isn't now in OpenVZ.

There is cfq scheduler in Linux kernel, that allows to assign IO priority to the process. It supports three classes:

real time (rt)
best effort (be)
idle class (i)

Within rt class and be class are 8 levels of priority. The more level is - more time for input/output particular process has. Additional information can be found for example at
http://www.mjmwired.net/kernel/Documentation/block/ioprio.tx t

So what do they do in Vserver?
When user sets certain IO priority to the context, Vserver framework just sets this IO priority to all processes in context! And that's all that they do, but this isn't right. Just look at these example:

1st context: 3 processes - priority be:4
2nd context: 1 process - priority be:6

So user expects that 2nd context has more IO bandwidth, but this isn't true, 'cause 1st context has more processes! And the more processes 1st context has more IO bandwidth it has.

Some time ago there were patches to do the same in OpenVZ, but do we need such implementation?
To create more sophisticated and true IO scheduling more investigation is necessary. Also there is also a big problem, 'cause pages can be written to the block device, when information about process isn't available any more...

HTH,
vass.

Subject: Re: IO scheduling
Posted by dagr on Thu, 21 Sep 2006 07:22:06 GMT
View Forum Message <> Reply to Message

tks for detailed explanation.

So do i get right , that dos attack is still quite possible ?
For instance if i have a quite big disk quota for VPS - say half of phys disk. I can fill it with a file and start a benchmark like iometer for random access pattern. BTW - does virtuozzo have extra features in this terms ?

Subject: Re: IO scheduling
Posted by Vasily Tarasov on Thu, 21 Sep 2006 08:03:19 GMT
View Forum Message <> Reply to Message

Sorry, I don't really sure that dos attack and IO are closely connected. I believe that most dos attacks are based on have network traffic (this isn't block I/O)...

Quote:For instance if i have a quite big disk quota for VPS - say half of phys disk. I can fill it with a file and start a benchmark like iometer for random access pattern.

Ahh... I suppose I get now what you mean saying "dos" attack in this situation! Well, yes, there is no possibility to directly limit IO now, not in OpenVZ not in Vserver

Subject: Re: IO scheduling
Posted by HaroldB on Mon, 25 Sep 2006 07:05:04 GMT
View Forum Message <> Reply to Message

Hello. The ability for a VE to utilize all of the disk i/o bandwidth in a system seems to be a very big problem. Has anyone investigated a project called CKRM?

"If you want a way to assign io priorities without relying on process inheritance and (re)nice you might find CKRM, with it's cfq-based IO controller, useful.

Quote:
Basically you create a set of classes that group tasks and give an
appropriate share of IO performance to tasks in that class. As processes get created CKRM will assign tasks to the IO classes based on a set of rules."

ref:

http://ckrm.sourceforge.net/
http://www.gatago.com/linux/kernel/14683383.html

Seems like if each CKRM "class" was a openvz VE, this could be a nice framework for limiting and more importantly guaranteeing disk i/o bandwidth per VE. Quoted from the CKRM patch:

Quote:
Resource allocations for a class is controlled by the parameters:

guarantee: specifies how much of a resource is guranteed to a class. A
special value DONT_CARE(-2) mean that there is no specific
guarantee of a resource is specified, this class may not get
any resource if the system is runing short of resources

limit: specifies the maximum amount of resource that is allowed to be
allocated by a class. A special value DONT_CARE(-2) mean that
there is no specific limit is specified, this class can get all
the resources available.

total_guarantee: total guarantee that is allowed among the children of this
class. In other words, the sum of "guarantee"s of all children
of this class cannot exit this number.

max_limit: Maximum "limit" allowed for any of this class's children. In
other words, "limit" of any children of this class cannot exceed
this value.

Subject: Re: IO scheduling
Posted by wfischer on Wed, 27 Sep 2006 11:04:51 GMT
View Forum Message <> Reply to Message

Another question regarding io scheduler:

To me it seems that OpenVZ uses anticipatory io scheduler as default io scheduler, according to the info in /var/log/messages after booting OpenVZ kernel (on a CentOS 4.4 host):
Sep 22 09:14:04 wc1 kernel: Using anticipatory io scheduler

I'm not an expert on io scheduling, but I heard that anticipatory io scheduler is mainly useful for desktop machines and for servers deadline or cfq schedulers should be used.

Could you give a short explanation why anticipatory io scheduler is used as default io scheduler in OpenVZ?

Thanks,
best wishes,
Werner

added remark: I just noticed that the anticipatory io scheduler has different default values in OpenVZ than in vanilla kernel (read_expire 10 instead of 125, read_batch_expire 10 instead of 500) - according to the suggestion on http://bugzilla.kernel.org/show_bug.cgi?id=5900#c1

Subject: Re: *SOLVED* IO scheduling
Posted by HaroldB on Wed, 27 Sep 2006 11:29:40 GMT
View Forum Message <> Reply to Message

What kernel version do you see anticipatory as the default scheduler? CFQ should be. CentOS 4.4 tells us almost nothing, tell us the kernel version.

Subject: Re: IO scheduling
Posted by Vasily Tarasov on Wed, 27 Sep 2006 11:41:27 GMT
View Forum Message <> Reply to Message

Quote:Could you give a short explanation why anticipatory io scheduler is used as default io scheduler in OpenVZ?

Default OpenVZ IO-scheduler is inherited from kernel on which particular OpenVZ kernel is based. For example:
1. 2.6.16-026test* and 2.6.8-022stab* are based on appropriate vanilla kernel, that use anticipatory scheduler as default (only since recent 2.6.18 vanilla kernel uses CFQ). So these OpenVZ kernels use anticipatory scheduler.
2. 2.6.9-023stab* is based on rhel4 kernel, that uses cfq as default, hence OpenVZ uses this scheduler.
...

I don't think this is fundamentally what is default IO scheduler:
you can change it any time to scheduler that best fits you!

HTH,
vass.

Subject: Re: IO scheduling
Posted by wfischer on Wed, 27 Sep 2006 12:17:36 GMT
View Forum Message <> Reply to Message

Thanks a lot for the info and the fast reply, it helps me a lot.
@HaroldB: I'm using OpenVZ Kernel 2.6.8-022stab078.14, so that is the reason why anticipatory is default according to the explanation from Vass.

best regards,
Werner