Subject: Re: first stable release of OpenVZ kernel virtualization solution
Posted by Ingo Molnar on Tue, 06 Dec 2005 14:04:16 GMT

View Forum Message <> Reply to Message

* Andrey Savochkin <saw@sawoct.com> wrote:

> > maybe i'm banging on open doors, but the same would be the case not only
> > for userspace-VM overcommit, but also for dirty data. I.e. there should
> > be (already is?) a per-instance 'dirty data threshold', to not force
> > other instances into waiting for writeout/swapout to happen.
>
> OVZ certainly has room for improvements with respect to swap. What I
> want to point out is that swapout management is a complex task. When a
> low-priority VPS exceeds its limits, it is not always benefitial for
> others to make it swap out: swapout wastes disk bandwidth, and to some
> extent CPU power. 'Dirty data threshold' could have helped, but it
> reduces the overall performance of the system, especially if the
> number of VPSs is small. Imagine only one VPS running: artificial
> 'dirty data threshold' would certainly be counter-productive.

but what you have right now is an in essence swapless system, correct?
Do you support swapping at all in OVZ instances?

Ingo

Subject: Re: first stable release of OpenVZ kernel virtualization solution
Posted by dev on Tue, 06 Dec 2005 14:16:33 GMT

View Forum Message <> Reply to Message

Ingo Molnar wrote:
> * Andrey Savochkin <saw@sawoct.com> wrote:
>>OVZ certainly has room for improvements with respect to swap. What I
>>want to point out is that swapout management is a complex task. When a
>>low-priority VPS exceeds its limits, it is not always benefitial for
>>others to make it swap out: swapout wastes disk bandwidth, and to some
>>extent CPU power. 'Dirty data threshold' could have helped, but it
>>reduces the overall performance of the system, especially if the
>>number of VPSs is small. Imagine only one VPS running: artificial
>>'dirty data threshold' would certainly be counter-productive.
>
> but what you have right now is an in essence swapless system, correct?
> Do you support swapping at all in OVZ instances?

Yes, swap is supported and processes are swapped in/out as in usual
kernel. The only difference in comparison with std kernel is that UBC
limits the amount of swappable memory VPS can have.

Kirill

Subject: Re: first stable release of OpenVZ kernel virtualization solution
Posted by Ingo Molnar on Tue, 06 Dec 2005 14:22:15 GMT
View Forum Message <> Reply to Message

* Kirill Korotaev <dev@sw.ru> wrote:

> >but what you have right now is an in essence swapless system, correct?
> >Do you support swapping at all in OVZ instances?

> Yes, swap is supported and processes are swapped in/out as in usual
> kernel. The only difference in comparison with std kernel is that UBC
> limits the amount of swappable memory VPS can have.

i mean, the only way to protect a high-prio instance against a low-prio instance doing heavy swapout is by making the low-prio instance swapless, correct? (either by not enabling it to swap at all, or by tweaking the UBC limits in a way that can never lead to swapping).

how about the 'dirty data creator' scenario: an instance filling up all of the RAM with dirty data, at which point a highprio instance is significantly impacted.

my point is, that such a swap or writeout related slowdown of a highprio instance can be just as bad as a real DoS - and it brings us essentially back to where we started with vserver. (and writeout related slowdowns of unrelated instances cannot be avoided even with the most conservative UBC settings, correct?)

Ingo

Subject: Re: first stable release of OpenVZ kernel virtualization solution
Posted by dev on Tue, 06 Dec 2005 15:48:25 GMT
View Forum Message <> Reply to Message

> * Kirill Korotaev <dev@sw.ru> wrote:
>>>but what you have right now is an in essence swapless system, correct?
>>>Do you support swapping at all in OVZ instances?
>
>>Yes, swap is supported and processes are swapped in/out as in usual
>>kernel. The only difference in comparison with std kernel is that UBC
>>limits the amount of swappable memory VPS can have.
>
> i mean, the only way to protect a high-prio instance against a low-prio
> instance doing heavy swapout is by making the low-prio instance
> swapless, correct? (either by not enabling it to swap at all, or by
> tweaking the UBC limits in a way that can never lead to swapping).
correct. But not active/big swapping is ok, as it usually leads system

to some equlibrium... only swap hog is bad.

> how about the 'dirty data creator' scenario: an instance filling up all
> of the RAM with dirty data, at which point a highprio instance is
> significantly impacted.
yes, this can be a problem which should be solved yet.
This can also be limited by UBC settings, but in general your point is
valid.

> my point is, that such a swap or writeout related slowdown of a highprio
> instance can be just as bad as a real DoS - and it brings us essentially
> back to where we started with vserver. (and writeout related slowdowns
> of unrelated instances cannot be avoided even with the most conservative
> UBC settings, correct?)
We plan to use CFQv2 in some near future, but currently writeout is not
controlled by UBC anyhow.
The only note is that currently used disk I/O scheduler (anticipatory)
behaves quite well when one VPS is doing massive writes...
Disk I/O is a kind of problem for any of existing virtualization
solutions and OpenVZ is not different here...

Kirill

Subject: Re: first stable release of OpenVZ kernel virtualization solution
Posted by Ingo Molnar on Tue, 06 Dec 2005 17:12:58 GMT

View Forum Message <> Reply to Message

* Kirill Korotaev <dev@sw.ru> wrote:

> >my point is, that such a swap or writeout related slowdown of a highprio
> >instance can be just as bad as a real DoS - and it brings us essentially
> >back to where we started with vserver. (and writeout related slowdowns
> >of unrelated instances cannot be avoided even with the most conservative
> >UBC settings, correct?)

> We plan to use CFQv2 in some near future, but currently writeout is
> not controlled by UBC anyhow. The only note is that currently used
> disk I/O scheduler (anticipatory) behaves quite well when one VPS is
> doing massive writes... Disk I/O is a kind of problem for any of
> existing virtualization solutions and OpenVZ is not different here...

it's not just massive disk IO and IO starvation of another instance
(which can be mitigated by isolating the disks of instances), it's also
the starvation of RAM of another instance.

Or is that solved already? I.e. can high-prio instances have a
guaranteed amount of RAM set aside for them - even if they do not make

use of that guaranteed amount at the moment?

this is where the Xen approach still seems to differ so much: there the
RAM assigned to an instance truly belongs to that instance. I.e. you can
achieve guaranteed service, no matter what another instance does.

Ingo

Subject: Re: first stable release of OpenVZ kernel virtualization solution
Posted by dev on Tue, 06 Dec 2005 19:44:21 GMT

View Forum Message <> Reply to Message

> * Kirill Korotaev <dev@sw.ru> wrote:
>
>
>>>my point is, that such a swap or writeout related slowdown of a highprio
>>>instance can be just as bad as a real DoS - and it brings us essentially
>>>back to where we started with vserver. (and writeout related slowdowns
>>>of unrelated instances cannot be avoided even with the most conservative
>>>UBC settings, correct?)
>
>
>>We plan to use CFQv2 in some near future, but currently writeout is
>>not controlled by UBC anyhow. The only note is that currently used
>>disk I/O scheduler (anticipatory) behaves quite well when one VPS is
>>doing massive writes... Disk I/O is a kind of problem for any of
>>existing virtualization solutions and OpenVZ is not different here...
>
>
> it's not just massive disk IO and IO starvation of another instance
> (which can be mitigated by isolating the disks of instances), it's also
> the starvation of RAM of another instance.
>
> Or is that solved already? I.e. can high-prio instances have a
> guaranteed amount of RAM set aside for them - even if they do not make
> use of that guaranteed amount at the moment?

Ah, I see your question now. Right now it is guaranteed only if no
overcommitment on the node, i.e. we have no guarantees in UBCs. But
actually I suppose it is not hard to implement such guarantees for
memory via page pools for VPSs which require such guarantees.
There were no real demands or at least we never heard such.

> this is where the Xen approach still seems to differ so much: there the
> RAM assigned to an instance truly belongs to that instance. I.e. you can
> achieve guaranteed service, no matter what another instance does.

sure, the approach is different. Our goals were bit different - to get
the highest VPS density.

BTW, AFAIK Xen is trying to implement so-called "balooning", which actually is a direction to OpenVZ approach, when memory is not locked behind the VPS.

Kirill