
Subject: vswap question - physpages/oomguarpages
Posted by [Rene Dokbua](#) on Thu, 21 Jun 2012 04:44:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

I just noticed a couple of containers on one of our vswap enabled servers have non-zero failcnt's (user_beancounters)

uid	resource	held	maxheld
barrier	limit	failcnt	
1413:			
	physpages	84873	131166
0	131072	61	
	oomguarpages	94628	377952
0	9223372036854775807		4
1409:			
	physpages	52986	262155
0	262144	1378	
	oomguarpages	57155	376725
0	9223372036854775807		18

(I've deleted lines with zero failcnt for clarity)

It was my understanding that if a vswap enabled container try to use more physpages than available it would start "vswapping" - is that what happens here? Otherwise how can physpages fail? Is this normal swapping behavior or a problem?

A second thing is, when I tried to raise the physpages value with vzctl it again wrote the old "block" values into the vz conf files, i.e.

i.e. before running vzctl the values are like this:

```
< PHYSPAGES="0:512M"  
< SWAPPAGES="0:1024M"
```

after running vzctl --set 1413 --save --ram 512M --swap 1G the values are like this:

```
> PHYSPAGES="0:131072"  
> SWAPPAGES="0:262144"
```

I already opened a bug report on this and I thought it had been fixed already. Is there some undocumented flag that needs to be provided to vzctl to get it to write out the values in the easy human readable format?

Regards,
Rene

Subject: Re: vswap question - physpages/oomguarpages
Posted by [Kirill Korotaev](#) on Thu, 21 Jun 2012 05:37:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

I guess it means oom, i.e. out of memory killer had to kill someone since vswap+ram was not enough...

Sent from my iPhonespam SPAMSPAM

On 21.06.2012, at 8:58, "Rene C." <openvz@dokbua.com> wrote:

>
> I just noticed a couple of containers on one of our vswap enabled servers have non-zero failcnt's (user_beancounters)
>
> uid resource held maxheld barrier limit
failcnt
> 1413:
> physpages 84873 131166 0 131072
61
> oomguarpages 94628 377952 0 9223372036854775807
4
> 1409:
> physpages 52986 262155 0 262144
1378
> oomguarpages 57155 376725 0 9223372036854775807
18
>
> (I've deleted lines with zero failcnt for clarity)
>
> It was my understanding that if a vswap enabled container try to use more physpages than available it would start "vswapping" - is that what happens here? Otherwise how can physpages fail? Is this normal swapping behavior or a problem?
>
> A second thing is, when I tried to raise the physpages value with vzctl it again wrote the old "block" values into the vz conf files, i.e.
>
> i.e. before running vzctl the values are like this:
> < PHYSPAGES="0:512M"
> < SWAPPAGES="0:1024M"
>
> after running vzctl --set 1413 --save --ram 512M --swap 1G the values are like this:
> > PHYSPAGES="0:131072"
> > SWAPPAGES="0:262144"
>
> I already opened a bug report on this and I though it had been fixed already. Is there some undocumented flag that needs to be provided to vzctl to get it to write out the values in the easy human readable format?
>

> Regards,
> Rene
>
> <ATT00001.c>

Subject: Re: vswap question - physpages/oomguarpages
Posted by [kir](#) on Tue, 26 Jun 2012 19:16:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 06/21/2012 08:44 AM, Rene C. wrote:

> A second thing is, when I tried to raise the physpages value with
> vzctl it again wrote the old "block" values into the vz conf files, i.e.
>
> i.e. before running vzctl the values are like this:
> < PHYSPAGES="0:512M"
> < SWAPPAGES="0:1024M"
>
> after running vzctl --set 1413 --save --ram 512M --swap 1G the values
> are like this:
> > PHYSPAGES="0:131072"
> > SWAPPAGES="0:262144"
>
> I already opened a bug report on this and I thought it had been fixed
> already.

A bug report is a bug report. It have status, and whenever something about the bug changes, it is usually reflected in the bug report.

> Is there some undocumented flag that needs to be provided to vzctl to
> get it to write out the values in the easy human readable format?

I have replied in that bug report. vzctl was never doing this (ie it was always saving parameters in simple numeric form, without any suffixes). More to day, doing it right requires non-trivial changes, so I set this bug severity to be enhancement.
