
Subject: How is the privvmpages limit reached?

Posted by [Wheel](#) on Sat, 09 Feb 2008 12:45:57 GMT

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

I'm trying to run Java (memory allocation intensive) on a VPS, and getting privvmpages limit problems.

I know that increasing privvmpages limit, using Xen, or decreasing the Java XMX usage, will fix the problem. But I'm wondering why and how Java reaches this privvmpages limit.

I have two VPSes:

XEN VPS:

-256mb guaranteed

-512mb swap

-No memory problems

```
[root@vps proc]#ps -aux --sort -vsize
```

```
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
```

```
USER    PID %CPU %MEM  VSZ  RSS TTY   STAT START  TIME COMMAND
tomcat  18099 0.2 26.9 236172 70580 ?    SI  Feb08  3:39 jsvc.exec -user tomcat -home
/usr/java/default -Dcatalina.home=/opt/tomcat -Dcatalina.base=/opt/
tomcat  23131 27.7 16.9 222084 44592 pts/2  SI+ 10:19  0:09 /usr/java/default/bin/java
-classpath /usr/share/java/ant.jar:/usr/share/java/ant-launcher.jar:/
tomcat  23091 25.4 17.0 221920 44692 pts/0  SI+ 10:19  0:09 /usr/java/default/bin/java
-classpath /usr/share/java/ant.jar:/usr/share/java/ant-launcher.jar:/
tomcat  23306 56.9 16.7 220756 44052 pts/1  SI+ 10:19  0:08 /usr/java/default/bin/java
-classpath /usr/share/java/ant.jar:/usr/share/java/ant-launcher.jar:/
mysql   18071 0.4 12.3 147652 32312 ?    SI  Feb08  6:30 /usr/libexec/mysqld
--defaults-file=/etc/my.cnf --basedir=/usr --datadir=/var/lib/mysql --user=m
```

```
.....
```

```
[root@vps proc]#free
```

```
      total    used    free   shared  buffers   cached
Mem:    262312  259472    2840        0     1216    35272
-/+ buffers/cache:  222984    39328
Swap:   525304    2772   522532
```

OpenVZ VPS:

-384MB guaranteed

- ~800MB privvmpages/burst?

-Memory problems when using the same apps as the XEN vps.

```
[root@vps ]# ant -f build-run-linux.xml
```

```
Error occurred during initialization of VM
```

```
Could not reserve enough space for code cache
```

```
[root@vps proc]# ps -aux --sort -vsize
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
USER    PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
tomcat  17460 0.3  6.1 203984 48340 ?        SI   01:44   0:06 jsvc.exec -user tomcat -home
/usr/java/default -Dcatalina.home=/opt/tomcat -Dcatalina.base=/opt/tomcat
-Djava.io.tmpdir=/var/tmp -w
root    23818 71.3  5.0 193520 39484 pts/1    SI+  02:14   0:04 /usr/java/latest/jre/bin/java
-classpath /usr/share/ant/lib/ant-launcher.jar -Dant.home=/usr/share/ant
-Dant.library.dir=/usr/share
root    23808 56.0  5.0 193348 40052 pts/0    SI+  02:14   0:04 /usr/java/latest/jre/bin/java
-classpath /usr/share/ant/lib/ant-launcher.jar -Dant.home=/usr/share/ant
-Dant.library.dir=/usr/share
mysql   16196 0.1  3.8 143256 30148 pts/0    SI   01:41   0:02 /usr/libexec/mysqld
--defaults-file=/etc/my.cnf --basedir=/usr --datadir=/var/lib/mysql --user=mysql
--pid-file=/var/run/mysqld/mys
.....
```

```
[root@vps proc]# cat user_beancounters
```

```
Version: 2.5
```

uid	resource	held	maxheld	barrier	limit	failcnt
982850:	kmemsize	7287695	7764715	2147483646		2147483646
	0					
	lockedpages	0	0	999999	999999	0
	privvmpages	188389	191204	196608	196608	
3						
	shmpages	3885	3901	98304	98304	
0						
	dummy	0	0	0	0	0
	numproc	109	113	999999	999999	0
	physpages	40864	43050	0	2147483647	
0						
	vmguarpages	0	0	98304	2147483647	
0						
	oomguarpages	40864	43050	98304	2147483647	
0						
	numtcpsock	62	67	7999992	7999992	
0						
	numflock	6	11	999999	999999	0
	numpty	4	4	500000	500000	0
	numsignifo	0	2	999999	999999	0
	tcpsndbuf	1292928	1334288	80530432	262556672	
0						
	tcprcvbuf	1249536	1371040	80530432	262556672	

0	othersockbuf	13920	32896	80530432	262556672	
0	dgramrcvbuf	0	8464	80530432	262556672	
0	numothersock	18	24	7999992	7999992	
0	dcachesize	0	0	2147483646	2147483646	
0	numfile	2571	2638	23999976	23999976	
0	dummy	0	0	0	0	0
	dummy	0	0	0	0	0
	dummy	0	0	0	0	0
	numiptent	14	14	999999	999999	0

```
[root@vps proc]# free
      total    used    free   shared  buffers   cached
Mem:   786432  755120   31312     0        0        0
-/+ buffers/cache: 755120  31312
Swap:      0        0        0
```

As can be seen above, the XEN vps with less memory than the OpenVZ vps, did not cause any memory problems, while the OpenVZ vps with more total (inc burst) memory did cause memory problems by hitting privvmpages.

I read on a number of different sites that hitting privvmpages is caused by OpenVZ showing the total amount of RAM on the hardware machine/Host instead of the amount of RAM on the VPS.

Someone said to me "OpenVZ does offer SLM like propoerties in reporting RAM so you should be ok with day to day usage of it rather than the old method where is reported the RAM of the host". And the "free" command also shows a total RAM of the VPS and not that of the host.

My question is:

- 1) Why is the OpenVZ having memory problems while it has more memory than the Xen VPS?
- 2) Is the statement correct that Java sees [as total ram] the amount of the VPS (~800mb) and not that of the host?

Subject: Re: How is the privvmpages limit reached?
 Posted by [rickb](#) on Sat, 09 Feb 2008 16:48:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi, it sounds like you have been reading webhostingtalk for technical information. not a good idea..

jre is known for allocating far more memory than it needs. due to this, your privvmpages will be much higher than physpages. other applications such as bind exhibit this as well. this is why you will never see privvmpages.limit=vmguarpages.limit

@@@

privvm=188389

phys=40864

In this case, your apps are allocating X amounts of memory and using X/4 to store data in it. There is no way to correct this in openvz; you should provide an adequate privvmpages limit for your application's hunger. Privvmpages is not a "burst" memory in the true sense, as your applications cannot realistically store 196608 pages (your limit) of data in memory. but, a high privvmpages on a non overcommitted hardware node can allow overuse of your vmguarpages.limit, which is a memory burst. the amount of burstable memory is difficult to report, rather you can report the memory allocation burst. the non vz linux kernel allows you to allocate as much memory as you wish, which is why your xen environment does not have this problem.

Quote:Someone said to me "OpenVZ does offer SLM like properties in reporting RAM so you should be ok with day to day usage of it rather than the old method where is reported the RAM of the host"

there are 2 sources for memory reporting- proc/meminfo and proc/user_beancounters. the most accurate source is always user_beancounters.

Quote:And the "free" command also shows a total RAM of the VPS and not that of the host

read the vzctl man page for the 'meminfo' option; it is possible to customize proc/meminfo reporting.

Cheers

Rick

Subject: Re: How is the privvmpages limit reached?
Posted by [Wheel](#) on Sun, 10 Feb 2008 01:58:01 GMT

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

Hi,

Thank you for your reply.

I read your post, but I do yet not fully understand why the Xen vps doesn't have memory problems.

(Correct me if I'm wrong.)

On the XEN vps it is able to allocate at most 768mb (guaranteed+swap). And on the OpenVZ vps it is able to allocate at least 403mb (98304 vmguarpages barrier) and at most 805mb (196608 privvmpages barrier).

To me it looks like the OpenVZ did not yet reach the privvmpages barrier, (since has a higher privvmpages barrier than the unfailed Xen VPS), but it did fail.

The doc says:

Quote:If the current amount of allocated memory space exceeds the guarantee but below the barrier of privvmpages, allocations may or may not succeed, depending on the total amount of available memory in the system.

So can I conclude that the OpenVZ space has too little available memory in the hardware node? Or might there also be another reason?

Subject: Re: How is the privvmpages limit reached?

Posted by [rickb](#) on Sun, 10 Feb 2008 03:00:13 GMT

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

You do not see this problem on xen for the same reason you do not see it on a stock linux kernel.

<http://www.redhat.com/magazine/001nov04/features/vm/>

Quote:overcommit_memory is a value which sets the general kernel policy toward granting memory allocations. If the value is 0, then the kernel checks to determine if there is enough memory free to grant a memory request to a malloc call from an application. If there is enough memory, then the request is granted. Otherwise, it is denied and an error code is returned to the application. If the setting in this file is 1, the kernel allows all memory allocations, regardless of the current memory allocation state. If the value is set to 2, then the kernel grants allocations above the amount of physical RAM and swap in the system as defined by the overcommit_ratio value. Enabling this feature can be somewhat helpful in environments which allocate large amounts of memory expecting worst case scenarios but do not use it all.

<http://www.gnu.org/software/gnusound/Documentation/ar01s05.html>

Quote:Memory overcommit is a Linux kernel feature that lets applications allocate more memory than is actually available. The idea behind this feature is that some applications allocate large amounts of memory "just in case", but never actually use it. Thus, memory overcommit allows you to run more applications than actually fit in your memory, provided the applications don't actually use the memory they've allocated. If they do, then the kernel terminates the application.

GNUsound needs enough memory (RAM + swap) to load a file into memory completely. GNUsound will try to recover gracefully from memory allocation failures, but sometimes it simply can't. In particular, you may have problems when using a kernel that has memory overcommit enabled. This may result in GNUsound being killed as it tries to load the file. To try and solve the problem, you can either increase the amount of memory (by adding RAM or swap), or you can disable memory overcommit by typing (as root):

```
$ echo 2 > /proc/sys/vm/overcommit_memory
```

Subject: Re: How is the privvmpages limit reached?
Posted by [Wheel](#) on Sun, 10 Feb 2008 03:11:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

And is the conclusion from my previous post "that the OpenVZ space has too little available memory in the hardware node", correct?

Subject: Re: How is the privvmpages limit reached?
Posted by [rickb](#) on Sun, 10 Feb 2008 03:19:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

no. there is no realistic limitation on allocation except for privvmpages. I guess your question is why maxheld!=barrier. for a simple example with small numbers- if your barrier is 100, your maxheld 80, and you try to allocate 40- the allocation will fail and your maxheld is still 80.

Rick

Subject: Re: How is the privvmpages limit reached?
Posted by [xemul](#) on Fri, 15 Feb 2008 09:15:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Privvmpages accounting works like this:

There are two kinds of mappings

1. the ones that are backed by some file on disk, i.e. MAP_SHARED and !MAP_ANON
2. the ones, that will go to swap on memory shortage, i.e. MAP_PRIVATE or MAP_ANON

The privvmpages limits the total length of mappings of the 2nd type only within the ve. This is OK, since on memory shortage pages of the 1st type will go to disk in existing area (which is limited by vzquota by the way)

So privvmpages limit can be exceeded easily.
