Subject: 2GB Memory Limit per process (32bit)?
Posted by jacobwm on Thu, 22 Jun 2006 22:24:02 GMT
View Forum Message <> Reply to Message

I'm running the SMP version of the latest stable kernel (2.6., and it seems I cannot run a java process with a 2GB heap requirement. I can set it to 1800M (min/max), but not 2000M.

Is there an inherent memory limit for an individual process and if so is there a way to overcome this with the openvz (32-bit)? Would the 64-bit version work?

I tried the enterprise version and the 2.6.16 branch -- neither worked. The stock RHEL4 kernels work fine at allowing at least 2GB (haven't tried higher yet).

Unfortunately this particular java app caches a large amount of data and actually does need around 2GB or memory and will likely need more in the future.

BTW I set the privvmpages=3500M:3500M for this test. I also looked at the kernel config file, but didn't see anything specific to individual process limits (ulimit didn't help either), but I could be missing something.

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by Vasily Tarasov on Fri, 23 Jun 2006 06:45:16 GMT
View Forum Message <> Reply to Message

AFAIK, you can have up to 3GB memory on ia32 for a process...

Please, can you do:
cat /proc/user_beancounters
after start of this applicationr with 2GB heap.

I suppose there are some hitted limits, for example,
kmemsize.

Thanks!

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by jacobwm on Fri, 23 Jun 2006 14:02:47 GMT
View Forum Message <> Reply to Message

It definitely hits the privvmpages limit, but I've set it to 3500M:3500M and it still won't start with just 2000M on the guest process.

It also will not run on the host node, so I'm guessing there's something limiting it in the kernel.

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by dev on Fri, 23 Jun 2006 19:10:44 GMT
View Forum Message <> Reply to Message

does it hit privvmpages limit even when run in the host?
it should perfectly run and there should be no much difference in this regard between OVZ and RHEL4 kernel.

the only difference can be due to 4GB split option, but it should affect only RHEL4-enterprise kernel, though you are running SMP (according to the post).

So, maybe you can provide some hint on how it fails? is there something in dmesg? mmm... maybe it hits not only the privvmpages?
does your application require much time to fail? if not, you can try to strace it and find out on which operation it fails...

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by jacobwm on Fri, 23 Jun 2006 19:52:12 GMT
View Forum Message <> Reply to Message

Below is an strace snippet. Looks like the jvm is calling mmap2. The app doesn't even load, the JVM stops cold at startup.

I was looking at the wrong user bean counter... there are no failcnt's when this app doesn't start. I suspect it's because it doesn't actually try to allocate it but checks first to see if it is available?

```
stat64("/usr/java/j2sdk1.4.2_10/jre/lib/jsse.jar", {st_mode=S_IFREG|0644, st_size=902059, ...}) =
0
lstat64("/usr", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre/lib/jsse.jar", {st_mode=S_IFREG|0644, st_size=902059, ...}) =
0
open("/usr/java/j2sdk1.4.2_10/jre/lib/jsse.jar", O_RDONLY|O_LARGEFILE) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=902059, ...}) = 0
_llseek(3, 0, [902059], SEEK_END)       = 0
mmap2(NULL, 902059, PROT_READ, MAP_SHARED, 3, 0) = 0x4223f000
close(3)                                = 0
stat64("/usr/java/j2sdk1.4.2_10/jre/lib/jce.jar", {st_mode=S_IFREG|0444, st_size=69736, ...}) = 0
lstat64("/usr", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
```

```
lstat64("/usr/java/j2sdk1.4.2_10/jre/lib/jce.jar", {st_mode=S_IFREG|0444, st_size=69736, ...}) = 0
open("/usr/java/j2sdk1.4.2_10/jre/lib/jce.jar", O_RDONLY|O_LARGEFILE) = 3
fstat64(3, {st_mode=S_IFREG|0444, st_size=69736, ...}) = 0
_llseek(3, 0, [69736], SEEK_END)        = 0
mmap2(NULL, 69736, PROT_READ, MAP_SHARED, 3, 0) = 0x4231c000
close(3)                                = 0
stat64("/usr/java/j2sdk1.4.2_10/jre/lib/charsets.jar", {st_mode=S_IFREG|0644, st_size=5905940,
...}) = 0
lstat64("/usr", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat64("/usr/java/j2sdk1.4.2_10/jre/lib/charsets.jar", {st_mode=S_IFREG|0644, st_size=5905940,
...}) = 0
open("/usr/java/j2sdk1.4.2_10/jre/lib/charsets.jar", O_RDONLY|O_LARGEFILE) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=5905940, ...}) = 0
_llseek(3, 0, [5905940], SEEK_END)      = 0
mmap2(NULL, 5905940, PROT_READ, MAP_SHARED, 3, 0) = 0x4232e000
close(3)                                = 0
stat64("/usr/java/j2sdk1.4.2_10/jre/classes", 0xbfffbe9c) = -1 ENOENT (No such file or directory)
mmap2(NULL, 33554432, PROT_READ|PROT_WRITE|PROT_EXEC,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0) = 0x428d0000
mmap2(0x428d0000, 163840, PROT_READ|PROT_WRITE|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x428d0000
mmap2(NULL, 524288, PROT_READ|PROT_WRITE|PROT_EXEC,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0) = 0x448d0000
mmap2(0x448d0000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x448d0000
mmap2(NULL, 163840, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x44950000
gettimeofday({1151092094, 33823}, NULL) = 0
gettimeofday({1151092094, 34036}, NULL) = 0
gettimeofday({1151092094, 34241}, NULL) = 0
gettimeofday({1151092094, 34533}, NULL) = 0
gettimeofday({1151092094, 34692}, NULL) = 0
mmap2(NULL, 2164260864, PROT_READ|PROT_WRITE|PROT_EXEC,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0) = -1 ENOMEM (Cannot allocate
memory)
write(1, "Error occurred during initializa"..., 43Error occurred during initialization of VM
) = 43
write(1, "Could not reserve enough space f"..., 46Could not reserve enough space for object
heap) = 46
write(1, "\n", 1
)                       = 1
unlink("/tmp/hsperfdata_root/16064")    = 0
exit_group(1)                           = ?
Process 16064 detached
```

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by jacobwm on Fri, 23 Jun 2006 19:54:01 GMT
View Forum Message <> Reply to Message

Oh and here's the user bean counter

ersion: 2.5

uid	resource	held	maxheld	barrier	limit	failcnt
7001:	kmemsize	980916	1051883	104857600	104857600	0
	lockedpages	0	0	32	32	0
	privvmpages	4402	13695	896000	896000	0
	shmpages	39	39	8192	8192	0
	dummy	0	0	0	0	0
	numproc	19	21	65	65	0
	physpages	2486	3420	0	2147483647	0
	vmguarpages	0	0	6144	2147483647	0
	oomguarpages	2486	3420	6144	2147483647	0
	numtcpsock	3	3	80	80	0
	numflock	3	4	100	110	0
	numpty	1	1	16	16	0
	numsiginfo	0	2	256	256	0
	tcpsndbuf	0	0	319488	524288	0
	tcprcvbuf	0	0	319488	524288	0
	othersockbuf	2228	10456	132096	336896	0
	dgramrcvbuf	0	8372	132096	132096	0
	numothersock	5	17	80	80	0
	dcachesize	87822	101024	1048576	1097728	0
	numfile	217	238	2048	2048	0
	dummy	0	0	0	0	0
	dummy	0	0	0	0	0
	dummy	0	0	0	0	0
	numiptent	10	10	128	128	0

-bash-3.00#

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by dev on Sun, 25 Jun 2006 07:30:15 GMT
View Forum Message <> Reply to Message

ideas to check:
1. check that you have the same overcommit mode under both kernels:

cat /proc/sys/vm/overcommit_memory

2. it would be nice if you could catch /proc/<PID>/maps of this process before the last mmap. I guess there is simply no place in virtual address space where contigious 2GB can be allocated. The same would be interesting to see on RHEL.

Does it work on SMP or enterprise RHEL kernel? or both?

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by dev on Mon, 26 Jun 2006 08:18:18 GMT
View Forum Message <> Reply to Message

maybe, you can give me an access to check?

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by frankfischer on Mon, 26 Jun 2006 13:40:39 GMT
View Forum Message <> Reply to Message

I'm not 100% sure but I think youre problem is the jvm, not openVz nor linux. There is a limit of 2GB for the 32-bit JVM:

http://www.unixville.com/~moazam/categories/jvmInternals/
and here

http://www.theserverside.com/discussions/thread.tss?thread_i d=26347

Best regards,

Frank

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by jacobwm on Mon, 26 Jun 2006 14:47:02 GMT
View Forum Message <> Reply to Message

I guess my only problem with that is the same JVM runs fine in both the SMP and UP versions of RHEL4 kernels - stock. I haven't tried the enterprise version, no need for it just yet.

However, the articles do mention that the JVM needs a *contiguous* block, not so much just the available memory. That might be difficult to manage/pull off in VPS's without huge amounts of physical memory to begin with.

I suspect we're going to have to move to an all 64-bit platform soon anyway as this cache size is only going to grow and 32-bit addressing isn't going to suffice.

As for trying it out, I'm merely using the j2sdk-1_4_2_10-linux-i586.rpm and setting the min/max ($JAVA_HOME/bin/java -server -Xms2000m -Xmx2000m) and it poops out right away.

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by dev on Mon, 26 Jun 2006 14:56:58 GMT
View Forum Message <> Reply to Message

ok, we will try it tomorrow ourselfes if it reproducable so easy.

yes, if you have growing demands then better to use 64 bit.
32bit address space is too small.
I guess on RHEL it works because it does randomization of address space and probably a bit bigger contigious address range can be available.

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by frankfischer on Wed, 13 Sep 2006 08:32:09 GMT
View Forum Message <> Reply to Message

Hi dev,

what is the result of your investigation?

Regards,

Frank

Subject: Re: 2GB Memory Limit per process (32bit)?
Posted by jacobwm on Fri, 29 Sep 2006 18:28:10 GMT
View Forum Message <> Reply to Message

I'm not sure if the developers got anywhere, but we purchased 64-bit systems with 16GB of RAM. What I've noticed is there are still issues periodically with respect to contiguous memory being available on the system.

It seems to be more of a problem with how the contiguous memory as the VPS runs longer.

I would assume the kernel is managing the freeing up of memory but I don't know how it applies to the VPS as far as contiguous blocks. Not sure if there is a command i can issue anywhere that "defragments" memory. I might try http://ramdefrag.sourceforge.net/, even if it only can work on the HN, that would help.

Java just sucks for high memory requirements and doesn't seem to fit very well with VPS's because of it's insistance on contiguous chunks.

Subject: Re: 2GB Memory Limit per process (32bit)?

Posted by dev on Sat, 30 Sep 2006 11:26:57 GMT
View Forum Message <> Reply to Message

on x86-64 address space is really huge. So if you experience such problems even on x8664 it is more likely to be a bug.
Can you help with debugging the issue?