
Subject: Kernel cache dentry leak?

Posted by [insider](#) on Sun, 15 Jan 2012 17:56:49 GMT

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

Hello,

After setup node server with Centos 6 and 64bit latest openvz kernel 2.6.32-042stab044.11 we have noticed, that even on empty node after the two days node memory usage is shown 40-50%! After the inspection of running processes we have not noticed any process which uses significant amount of memory.

After the searching google we found that there may be kernel cache problem, so we executed command:

```
sync && echo 2 >/proc/sys/vm/drop_caches
```

And all used memory is freed.

The problem is, that these in cache used memory is reported as "used" memory, not a reusable cache or etc?

Does someone experience this kind of problem? If yes, maybe there is any solution?

Or maybe this is an incorrect memory usage reporting problem, or this is memory leak?

We have no such problem in all our centos 5 32bit kernel nodes. On Centos 5 32bit nodes there is no increasing memory usage (by cache or etc).

Also, on other forum we have an answer that someone else got this problem too:

Quote:We are experiencing the same thing.

Linux xxxxx.yyyyyy.com 2.6.32-042stab044.11 #1 SMP Wed Dec 14 16:02:00 MSK 2011 x86_64
x86_64 x86_64 GNU/Linux

Current RAM usage:

```
free -m
```

	total	used	free	shared	buffers	cached
Mem:	23971	13559	10411	0	330	12444
-/+ buffers/cache:		784	23187			
Swap:	26622	6	26616			

After sync:

```
free -m
```

	total	used	free	shared	buffers	cached
Mem:	23971	1068	22902	0	330	458
-/+ buffers/cache:		279	23691			
Swap:	26622	6	26616			

There is currently 1 VM running on this node. Why in the world would the server use 12 GB RAM as cache?

slabtop command shows big and increasing "dentry" usage.

I see by running "slabtop", that "dentry" value is very big and increasing continually. After cache clear it drops, but then increases and increases again and again.

For example, on our nominal loaded Centos 5 32 bit node there is around 700k objects in "dentry" and uses around 106MB.

In this time, on the empty node with Centos 6 64 bit kernel "dentry" holds 7000k objects (x10 more than on our loaded node!) and uses 1.5GB RAM and increases...

Thank you for any help and answers.

Subject: Re: Kernel cache dentry leak?

Posted by [insider](#) on Fri, 20 Jan 2012 19:10:13 GMT

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

After a last manual cache clear with echo 2 >/proc/sys/vm/drop_caches there a 2 days passed, "dentry" now holds 15174252 objects and uses 3372056K and keeps increasing...

slabtop command information:

```
Active / Total Objects (% used) : 15292469 / 15303649 (99.9%)
Active / Total Slabs (% used)   : 851679 / 851681 (100.0%)
Active / Total Caches (% used)  : 122 / 240 (50.8%)
Active / Total Size (% used)    : 3232528.04K / 3234571.11K (99.9%)
Minimum / Average / Maximum Object : 0.02K / 0.21K / 4096.00K
```

```

OBJS ACTIVE USE OBJ SIZE SLABS OBJ/SLAB CACHE SIZE NAME
15174252 15174204 99% 0.21K 843014 18 3372056K dentry <<=====!!!!
24790 24759 99% 0.10K 670 37 2680K buffer_head
20048 19762 98% 0.03K 179 112 716K size-32
12528 12373 98% 0.08K 261 48 1044K sysfs_dir_cache
10384 9963 95% 0.06K 176 59 704K size-64
6816 6777 99% 0.62K 1136 6 4544K inode_cache
4928 3144 63% 0.05K 64 77 256K anon_vma_chain
4921 4204 85% 0.20K 259 19 1036K vm_area_struct
4500 4425 98% 0.12K 150 30 600K size-128
3899 3866 99% 0.55K 557 7 2228K radix_tree_node
3411 3404 99% 1.05K 1137 3 4548K ext4_inode_cache
3372 3354 99% 0.83K 843 4 3372K ext3_inode_cache
3205 3184 99% 0.68K 641 5 2564K proc_inode_cache
2862 2694 94% 0.07K 54 53 216K Acpi-Operand
2695 1937 71% 0.05K 35 77 140K anon_vma
1940 1116 57% 0.19K 97 20 388K cred_jar
1740 1714 98% 1.00K 435 4 1740K size-1024
1620 1583 97% 0.19K 81 20 324K size-192
1440 1055 73% 0.25K 96 15 384K filp
1380 1332 96% 0.04K 15 92 60K Acpi-Namespace
1376 1286 93% 0.50K 172 8 688K size-512
945 904 95% 0.84K 105 9 840K shmem_inode_cache
612 571 93% 2.00K 306 2 1224K size-2048
540 345 63% 0.25K 36 15 144K size-256
510 240 47% 0.11K 15 34 60K task_delay_info
468 287 61% 0.31K 39 12 156K skbuff_head_cache

```

424	56	13%	0.06K	8	53	32K fs_cache
420	222	52%	0.12K	14	30	56K pid
308	298	96%	0.53K	44	7	176K idr_layer_cache
288	231	80%	1.00K	72	4	288K signal_cache
288	29	10%	0.08K	6	48	24K blkdev_ioc
288	256	88%	0.02K	2	144	8K dm_target_io
280	239	85%	0.19K	14	20	56K kmem_cache
280	54	19%	0.13K	10	28	40K cfq_io_context
276	62	22%	0.03K	3	92	12K size-32(UBC)
276	256	92%	0.04K	3	92	12K dm_io
270	233	86%	2.06K	90	3	720K sighand_cache
260	238	91%	2.75K	130	2	1040K task_struct
242	242	100%	4.00K	242	1	968K size-4096
240	182	75%	0.75K	48	5	192K sock_inode_cache
202	2	0%	0.02K	1	202	4K jbd2_revoke_table
202	4	1%	0.02K	1	202	4K revoke_table
187	54	28%	0.69K	17	11	136K files_cache
168	56	33%	0.27K	12	14	48K cfq_queue
162	55	33%	0.81K	18	9	144K task_xstate
159	18	11%	0.06K	3	53	12K size-64(UBC)
153	99	64%	0.81K	17	9	136K UNIX
144	32	22%	0.02K	1	144	4K jbd2_journal_handle
124	74	59%	1.00K	31	4	124K size-1024(UBC)
120	18	15%	0.19K	6	20	24K bio-0
120	45	37%	0.12K	4	30	16K inotify_inode_mark_entry

Is there a way to dump contents of dentry to a file, maybe to inspect and investigate, what this cache contains?

I have tried with "dd" copy from /dev/mem to a files, but it not allows to dump full kernel memory...

Maybe this problem is related to a filesystem?

We have mounted these filesystems:

/dev/md2 on / type ext4 (rw)

proc on /proc type proc (rw)

none on /dev/pts type devpts (rw,gid=5,mode=620)

/dev/md0 on /boot type ext3 (rw)

/dev/mapper/vg0-vz on /vz type ext3 (rw)

none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)

beancounter on /proc/vz/beancounter type cgroup (rw,name=beancounter)

container on /proc/vz/container type cgroup (rw,name=container)

fairsched on /proc/vz/fairsched type cgroup (rw,name=fairsched)

Searching for a solution or a way to investigate this problem, but still unsuccessful...

Upgraded kernel to 2.6.32-042stab044.17 #1 SMP Fri Jan 13 12:53:58 MSK 2012 x86_64 x86_64 x86_64 GNU/Linux, but this not solved dentry leak problem.

Any thoughts?

Does nobody else got this problem with RHEL 6 63bit 2.6.32 ?

Subject: Re: Kernel cache dentry leak?

Posted by [disaster](#) on Sun, 22 Jan 2012 20:24:33 GMT

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

just subscribing to this topic. Didn't found a special knob for it.

Subject: Re: Kernel cache dentry leak?

Posted by [insider](#) on Fri, 27 Jan 2012 10:42:10 GMT

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

Well, I think we have found the problem with this dentry cache leak.

We decided to completely disable traffic shaping in all our nodes (tc/HTB). And, after that we noticed that dentry cache is not increasing! To test this, we have enabled traffic shaping in one of our nodes and dentry starts increasing on this node again.

So, to solve this dentry memory leak problem, you should try to completely disable traffic shaping and look what happens.

It seems that dentry cache leak is somehow related to traffic shaping?!

Subject: Re: Kernel cache dentry leak?

Posted by [mustardman](#) on Sat, 03 Mar 2012 04:50:38 GMT

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

I have the same problem on 2 different nodes on latest stable kernel.

2.6.32-042stab049.6 x86_64

Turning off traffic shaping does not seem to help. Only thing that works is the `sync && echo 2 >/proc/sys/vm/drop_caches` every once in awhile.

Subject: Re: Kernel cache dentry leak?

Posted by [kir](#) on Thu, 27 Sep 2012 07:39:36 GMT

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

Guys

(1) This is a subject of bug #2143:

http://bugzilla.openvz.org/show_bug.cgi?id=2143

(2) Currently it is unclear if this is a bug or not.

(3) If you want to help investigate/resolve this one, please try to reproduce as asked in bug's comment #22 (http://bugzilla.openvz.org/show_bug.cgi?id=2143#c22) and report your results to that bug report (not here).

Thanks.

Subject: Re: Kernel cache dentry leak?

Posted by [khorenko](#) on Thu, 27 Sep 2012 12:15:43 GMT

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

Hi again.

I'm really glad that we did not get any proof of an improper Linux kernel caches handling (>4 months passed up to now).

This means that the kernel uses all RAM available on the node (and not used by applications) for various caches, in particular for dentry cache, but any moment an application requests some more memory, those caches are shrunk automatically and there is no memory leak in the system.

The absence of "free" memory on a node (in case there is a lot of "cached" memory) is not a problem, it's a proper system behavior, it's a great feature of the Linux kernel to use all RAM not used at the moment by applications to speed up the overall node performance by using various caches. And dentry cache is a very helpful cache - it does minimize the number of disk drive accesses (and disk access is much slower than memory access).

BTW, in this thread I saw that people use a "workaround" by dropping/flushing caches via "vm.drop_caches" sysctl.

Guys, by this action you get a lot of "free" RAM shown by "free" command, this is true. But now for example any file operation will require real slow disk I/O instead of quick data get from the cache in memory => the overall node performance is significantly decreased until cache is filled up again.

And if you perform cache drop periodically - you periodically decrease the overall node performance.

If you want to check the performance degradation caused by dropping caches, just perform the following simple test (just an example I've executed on my own node):

```
# echo 3 > /proc/sys/vm/drop_caches; \  
echo "Caches are flushed."; \  
echo "Checking how long does it take to list all files in /usr with empty cache."; \  
time ls -lR /usr >/dev/null; \  
echo "Checking how long does it take to list all files in /usr with filled cache."; \  
time ls -lR /usr >/dev/null;
```

Caches are flushed.

Checking how long does it take to list all files in /usr with empty cache.

```
real 0m7.511s
user 0m0.425s
sys 0m1.612s
```

Checking how long does it take to list all files in /usr with filled cache.

```
real 0m0.791s
user 0m0.254s
sys 0m0.534s
```

Hope that helps.

P.S.

Quote:(2) Currently it is unclear if this is a bug or not.

This is not a bug, the system just uses the memory in the most efficient way and is ready to provide memory to applications any moment.
