

---

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 prove of an improper Linux kernel caches handling (>4 months passed up to now).

This means that 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.

---