



Is it normal that this kernel param (fs.quota.syncs) increases also by system which does not use quota

🕒 Updated May 30 2014 at 7:30 PM

🚩 Issue

- In a customer's system, quota is not using it.
- However, when three sets of systems were compared, "fs.quota.syncs" had a difference.
- As a result, when the sync command was used also by the system which is not using quota, the value of this parameter increased.
- Is it normal that this kernel parameter (fs.quota.syncs) increases also by the system which does not use quota?

Environment

- Red Hat Enterprise Linux Version Number: 6
- Release Number: 6.2
- Architecture: x86_64
- Kernel Version: 2.6.32-220.4.2.el6.x86_64

✅ Resolution

- Yes, it's normal.
- proc value of fs.quota.syncs , /proc/sys/fs/quota/syncs (internally called as "dqstats.syncs") is increased everytime vfs_quota_sync() is called in the kernel

Diagnostic Steps

In linux-2.6.32-220.4.2.el6.x86_64/fs/quota/dquot.c

```
<...> 2528 { 2529 .ctl_name = FS_DQ_SYNCS, 2530 .procname = "syncls", 2531 .data
= &dqstats.syncls, <=== here represents syncls value. 2532 .maxlen = sizeof(int),
2533 .mode = 0444, 2534 .proc_handler = &proc_dointvec, 2535 }, 2536 #ifdef
CONFIG_PRINT_QUOTA_WARNING
```

In linux-2.6.32-220.4.2.el6.x86_64/fs/quota/dquot.c

```
<...> 539 540 int vfs_quota_sync(struct super_block *sb, int type) 541 { 542
struct list_head *dirty; 543 struct dquot *dqopt; 544 struct quota_info *dqopt
= sb_dqopt(sb); 545 int cnt; 546 547 mutex_lock(&dqopt->dqonoff_mutex); 548 for
(cnt = 0; cnt < MAXQUOTAS; cnt++) { 549 if (type != -1 && cnt != type) 550
continue; <...> 577 if ((cnt == type || type == -1) && sb_has_quota_active(sb,
cnt) 578 && info_dirty(&dqopt->info[cnt])) 579 sb->dq_op->write_info(sb, cnt);
580 spin_lock(&dq_list_lock); 581 dqstats.syncls++; <===== syncls
count increases when vfs_quota_sync() is called. 582
spin_unlock(&dq_list_lock); 583 mutex_unlock(&dqopt->dqonoff_mutex); 584 585
return 0; 586 } 587 EXPORT_SYMBOL(vfs_quota_sync); 588 <...> 2460 2461 const
struct quotactl_ops vfs_quotactl_ops = { 2462 .quota_on = vfs_quota_on, 2463
.quota_off = vfs_quota_off, 2464 .quota_sync = vfs_quota_sync, <=====
vfs_quota_sync is a member of vfs_quotactl_ops 2465 .get_info = vfs_get_dqinfo,
2466 .set_info = vfs_set_dqinfo, 2467 .get_dqblk = vfs_get_dqblk, 2468
.set_dqblk = vfs_set_dqblk 2469 };
```

For example, when I run sync on the RHEL6.4 box, we can see following behaviour:

```
# cat /etc/redhat-release Red Hat Enterprise Linux Server release 6.4
(Santiago) # uname -a Linux localhost 2.6.32-358.14.1.el6.x86_64 #1 SMP Mon Jun
17 15:54:20 EDT 2013 x86_64 x86_64 x86_64 GNU/Linux # sysctl -a|grep
fs.quota.syncls fs.quota.syncls = 0 # sync <==== run 'sync' # sysctl -a|grep
fs.quota.syncls fs.quota.syncls = 4 <==== fs.quota.syncls +4 # sync <==== run
'sync' # sysctl -a|grep fs.quota.syncls fs.quota.syncls = 8 <==== fs.quota.syncls
+4
```

So in this case, we can expect that `vfs_quota_sync` was called 4 times per 'sync'.

Actually we can see this behaviour with following systemtap script.

- Example: `vfs_quota_sync.tap`

```
probe kernel.function("vfs_quota_sync") { printf("\n\n\n%s[%d] called %s\n",
execname(), pid(), probefunc()) printf("\nprint_backtrace():\n")
print_backtrace() }
```

On one terminal window, we can write and run systemtap script with `stap` command, then run 'sync' command on another terminal window.

When we run 'sync' command once, we can see following backtrace from systemtap script.

```
# stap ./vfs_quota_sync.tap sync[20720] called vfs_quota_sync <=== run *once*.
print_backtrace(): 0xffffffff811e3090 : vfs_quota_sync+0x0/0x210 [kernel] <===
vfs_quota_sync() has been called and syncs value increased. 0xffffffff811b1d96
: __sync_filesystem+0x36/0x90 [kernel] 0xffffffff811b1ee8 :
sync_filesystems+0xf8/0x130 [kernel] 0xffffffff811b1f77 : sys_sync+0x17/0x40
[kernel] 0xffffffff8100b288 : tracesys+0xd9/0xde [kernel] sync[20720] called
vfs_quota_sync print_backtrace(): 0xffffffff811e3090 : vfs_quota_sync+0x0/0x210
[kernel] <=== vfs_quota_sync() has been called and syncs value increased.
0xffffffff811b1d96 : __sync_filesystem+0x36/0x90 [kernel] 0xffffffff811b1ee8 :
sync_filesystems+0xf8/0x130 [kernel] 0xffffffff811b1f77 : sys_sync+0x17/0x40
[kernel] 0xffffffff8100b288 : tracesys+0xd9/0xde [kernel] sync[20720] called
vfs_quota_sync print_backtrace(): 0xffffffff811e3090 : vfs_quota_sync+0x0/0x210
[kernel] <=== vfs_quota_sync() has been called and syncs value increased.
0xffffffff811e62fa : sync_quota_sb+0x3a/0x130 [kernel] 0xffffffff811b1dda :
__sync_filesystem+0x7a/0x90 [kernel] 0xffffffff811b1ee8 :
sync_filesystems+0xf8/0x130 [kernel] 0xffffffff811b1f81 : sys_sync+0x21/0x40
[kernel] 0xffffffff8100b288 : tracesys+0xd9/0xde [kernel] sync[20720] called
vfs_quota_sync print_backtrace(): 0xffffffff811e3090 : vfs_quota_sync+0x0/0x210
[kernel] <=== vfs_quota_sync() has been called and syncs value increased.
0xffffffff811e62fa : sync_quota_sb+0x3a/0x130 [kernel] 0xffffffff811b1dda :
__sync_filesystem+0x7a/0x90 [kernel] 0xffffffff811b1ee8 :
sync_filesystems+0xf8/0x130 [kernel] 0xffffffff811b1f81 : sys_sync+0x21/0x40
[kernel] 0xffffffff8100b288 : tracesys+0xd9/0xde [kernel]
```

This shows that `vfs_quota_sync` is called when running 'sync', this is the reason why/how syncs count increases by syncing.

Product(s)

[Red Hat Enterprise Linux \(https://access.redhat.com/search/#/?q=product:\("Red+Hat+Enterprise+Linux"\)\)](https://access.redhat.com/search/#/?q=product:()

Component [kernel \(https://access.redhat.com/search/#/?q=component:\("kernel"\)\)](https://access.redhat.com/search/#/?q=component:()

Category [Learn more \(https://access.redhat.com/search/#/?q=category:\("Learn+more"\)\)](https://access.redhat.com/search/#/?q=category:()

Tags [kernel systemtap \(https://access.redhat.com/search/#/?q=tag:\("kernel+systemtap"\)\)](https://access.redhat.com/search/#/?q=tag:()

Comments

Add new comment

[Formatting Help](#)

SUBMIT COMMENT

[Privacy Policy \(http://www.redhat.com/en/about/privacy-policy\)](http://www.redhat.com/en/about/privacy-policy)

[Customer Portal Terms of Use \(https://access.redhat.com/help/terms/\)](https://access.redhat.com/help/terms/)

[All Policies and Guidelines \(http://www.redhat.com/en/about/all-policies-guidelines\)](http://www.redhat.com/en/about/all-policies-guidelines)

Copyright © 2015 Red Hat, Inc.