

XATTR cache performance

See [MRP-57](#)

Current (gerrit) code has been tested with 2 nodes:

- panda (MDS and OSS), Intel Atom D525 1.8 GHz 4-way (2 core + HT)
- panda-notebook (client), Athlon NEO MV40 1.6 GHz 1-way

connected by Ethernet 1000 Mb/s, full duplex (according to ethtool)

The test includes creating 1000 files, setting EA user.attr to val for all these files, then dropping caches and querying user.attr from all these files (see detailed transcript below). The timings are as follows:

xattr cache	1st getfattr (+ other client caching)	2nd getfattr	3rd getfattr
on	5.739s	0.632s	0.633s
off	6.355s	2.779s	2.857s

xattr cache	1st setfattr	2nd setfattr	3rd setfattr
on	5.634s	4.412s	4.485s
off	4.467s	4.437s	4.348s

A small performance gain even for the first getfattr happens because in real life getfattr calls getxattr twice: first with 0 buffer size to get the attr value length, then with the buffer size according to the returned value. After the first call of getfattr everything is cached so the second call is almost for free!

The first setxattr is slow (5.6 seconds) because it separately enqueues the LCK_PW lock. As discussed with Alexey, this can be fixed in later versions. Later setxattr calls are as effective as without the xattr cache.

getfattr tests

```
[root@panda-notebook lustre]# touch {0..1000}
[root@panda-notebook lustre]# setfattr -n user.attr -v value *
[root@panda-notebook lustre]# echo 3 > /proc/sys/vm/drop_caches
```

```

[root@panda-notebook lustre]# time getfattr -n user.attr * > /dev/null

real    0m5.739s
user    0m0.018s
sys     0m1.065s
[root@panda-notebook lustre]# time getfattr -n user.attr * > /dev/null

real    0m0.632s
user    0m0.013s
sys     0m0.620s
[root@panda-notebook lustre]# time getfattr -n user.attr * > /dev/null

real    0m0.633s
user    0m0.008s
sys     0m0.625s
[root@panda-notebook lustre]# echo 0 >
/proc/fs/lustre/llite/lustre-*/xattr_cache
[root@panda-notebook lustre]# echo 3 > /proc/sys/vm/drop_caches
[root@panda-notebook lustre]# time getfattr -n user.attr * > /dev/null

real    0m6.355s
user    0m0.009s
sys     0m0.961s
[root@panda-notebook lustre]# time getfattr -n user.attr * > /dev/null

real    0m2.779s
user    0m0.013s
sys     0m0.740s
[root@panda-notebook lustre]# time getfattr -n user.attr * > /dev/null

real    0m2.857s
user    0m0.005s
sys     0m0.711s

```

setfattr tests

```

[root@panda-notebook lustre]# touch {0..1000}
[root@panda-notebook lustre]# time setfattr -n user.attr -v value *

real    0m5.634s
user    0m0.008s
sys     0m0.533s
[root@panda-notebook lustre]# time setfattr -n user.attr -v value *

real    0m4.412s
user    0m0.005s
sys     0m0.373s
[root@panda-notebook lustre]# time setfattr -n user.attr -v value *

real    0m4.485s
user    0m0.008s

```

```
sys      0m0.368s
[root@panda-notebook lustre]# rm -f *
[root@panda-notebook lustre]# touch {0..1000}
[root@panda-notebook lustre]# find /proc/fs -name xattr_cache
/proc/fs/lustre/llite/lustre-ffff880074c41800/xattr_cache
[root@panda-notebook lustre]# echo 0 >
/proc/fs/lustre/llite/lustre-ffff880074c4
1800/xattr_cache
[root@panda-notebook lustre]# time setfattr -n user.attr -v value *

real    0m4.467s
user    0m0.007s
sys     0m0.415s
[root@panda-notebook lustre]# time setfattr -n user.attr -v value *

real    0m4.437s
user    0m0.013s
sys     0m0.380s
[root@panda-notebook lustre]# time setfattr -n user.attr -v value *

real    0m4.348s
user    0m0.003s
sys     0m0.362s
```