

## LOCK STATISTICS

## - WHAT

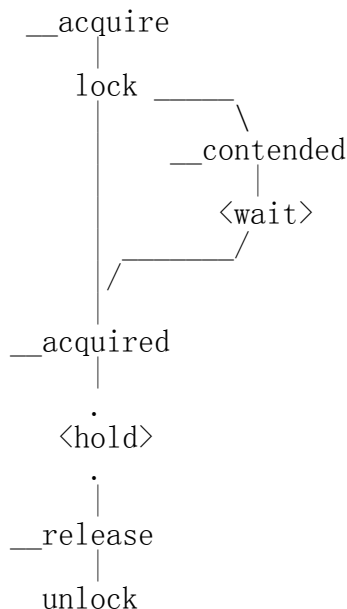
As the name suggests, it provides statistics on locks.

## - WHY

Because things like lock contention can severely impact performance.

## - HOW

Lockdep already has hooks in the lock functions and maps lock instances to lock classes. We build on that. The graph below shows the relation between the lock functions and the various hooks therein.



lock, unlock	- the regular lock functions
<u>*</u>	- the hooks
</>	- states

With these hooks we provide the following statistics:

con-bounces	- number of lock contention that involved x-cpu data
contentions	- number of lock acquisitions that had to wait
wait time min	- shortest (non-0) time we ever had to wait for a lock
max	- longest time we ever had to wait for a lock
total	- total time we spend waiting on this lock
acq-bounces	- number of lock acquisitions that involved x-cpu data
acquisitions	- number of times we took the lock
hold time min	- shortest (non-0) time we ever held the lock
max	- longest time we ever held the lock
total	- total time this lock was held

From these number various other statistics can be derived, such as:

lockstat.txt

hold time average = hold time total / acquisitions

These numbers are gathered per lock class, per read/write state (when applicable).

It also tracks 4 contention points per class. A contention point is a call site that had to wait on lock acquisition.

#### - CONFIGURATION

Lock statistics are enabled via CONFIG\_LOCK\_STATS.

#### - USAGE

Enable collection of statistics:

```
# echo 1 >/proc/sys/kernel/lock_stat
```

Disable collection of statistics:

```
# echo 0 >/proc/sys/kernel/lock_stat
```

Look at the current lock statistics:

( line numbers not part of actual output, done for clarity in the explanation below )

```
# less /proc/lock_stat
```

```
01 lock_stat version 0.3
02
```

```
-----
03                                     class name      con-bounces   contentions
waittime-min   waittime-max waittime-total  acq-bounces   acquisitions
holdtime-min   holdtime-max holdtime-total
04
-----
```

```
-----
05
06                                     &mm->mmap_sem-W:      233           538
18446744073708      22924.27      607243.51      1342          45806
    1.71           8595.89      1180582.34
07                                     &mm->mmap_sem-R:      205           587
18446744073708      28403.36      731975.00      1940          412426
    0.58           187825.45      6307502.88
08
09                                     &mm->mmap_sem      487
[<fffffffff8053491f>] do_page_fault+0x466/0x928
10                                     &mm->mmap_sem      179
[<fffffffff802a6200>] sys_mprotect+0xcd/0x21d
11                                     &mm->mmap_sem      279
[<fffffffff80210a57>] sys_mmap+0x75/0xce
12                                     &mm->mmap_sem      76
```

```

lockstat.txt
[<ffffffff802a490b>] sys_munmap+0x32/0x59
13
14          &mm->mmap_sem          270
[<ffffffff80210a57>] sys_mmap+0x75/0xce
15          &mm->mmap_sem          431
[<ffffffff8053491f>] do_page_fault+0x466/0x928
16          &mm->mmap_sem          138
[<ffffffff802a490b>] sys_munmap+0x32/0x59
17          &mm->mmap_sem          145
[<ffffffff802a6200>] sys_mprotect+0xcd/0x21d
18
19
.....
.....
.....
20
21          dcache_lock:          621          623
    0.52          118.26          1053.02          6745          91930
0.29          316.29          118423.41
22
23          dcache_lock          179
[<ffffffff80378274>] _atomic_dec_and_lock+0x34/0x54
24          dcache_lock          113
[<ffffffff802cc17b>] d_alloc+0x19a/0x1eb
25          dcache_lock          99
[<ffffffff802ca0dc>] d_rehash+0x1b/0x44
26          dcache_lock          104
[<ffffffff802cbca0>] d_instantiate+0x36/0x8a
27
28          dcache_lock          192
[<ffffffff80378274>] _atomic_dec_and_lock+0x34/0x54
29          dcache_lock          98
[<ffffffff802ca0dc>] d_rehash+0x1b/0x44
30          dcache_lock          72
[<ffffffff802cc17b>] d_alloc+0x19a/0x1eb
31          dcache_lock          112
[<ffffffff802cbca0>] d_instantiate+0x36/0x8a

```

This excerpt shows the first two lock class statistics. Line 01 shows the output version – each time the format changes this will be updated. Line 02-04 show the header with column descriptions. Lines 05-18 and 20-31 show the actual statistics. These statistics come in two parts; the actual stats separated by a short separator (line 08, 13) from the contention points.

The first lock (05-18) is a read/write lock, and shows two lines above the short separator. The contention points don't match the column descriptors, they have two: contentions and [<IP>] symbol. The second set of contention points are the points we're contending with.

The integer part of the time values is in us.

View the top contending locks:

```

# grep : /proc/lock_stat | head
          &inode->i_data.tree_lock-W:          15          21657
0.18          1093295.30 11547131054.85          58          10415          0.16

```

```

lockstat.txt
      87.51      6387.60
0.00      0.00      0.00      23302      0      231198      0      0.25
      8.45      98023.38
      dcache_lock:      1037      1161
0.38      45.32      774.51      6611      243371      0.15
      306.48      77387.24
      &inode->i_mutex:      161      286
18446744073709      62882.54      1244614.55      3653      20598
18446744073709      62318.60      1693822.74
      &zone->lru_lock:      94      94
0.53      7.33      92.10      4366      32690      0.29
      59.81      16350.06
      &inode->i_data.i_mmap_lock:      79      79
0.40      3.77      53.03      11779      87755      0.28
      116.93      29898.44
      &q->__queue_lock:      48      50
0.52      31.62      86.31      774      13131      0.17
      113.08      12277.52
      &rq->rq_lock_key:      43      47
0.74      68.50      170.63      3706      33929      0.22
      107.99      17460.62
      &rq->rq_lock_key#2:      39      46
0.75      6.68      49.03      2979      32292      0.17
      125.17      17137.63
      tasklist_lock-W:      15      15
1.45      10.87      32.70      1201      7390      0.58
      62.55      13648.47

```

Clear the statistics:

```
# echo 0 > /proc/lock_stat
```