

## FreeNAS - Bug #6521

### Dtrace is unhappy

11/03/2014 02:11 PM - Josh Paetzel

<b>Status:</b> Resolved	
<b>Priority:</b> Important	
<b>Assignee:</b> Jordan Hubbard	
<b>Category:</b>	
<b>Target version:</b> N/A	
<b>Seen in:</b> 9.3-RELEASE	<b>Needs Merging:</b> Yes
<b>Severity:</b> New	<b>Needs Automation:</b> No
<b>Reason for Closing:</b>	<b>Support Suite Ticket:</b> n/a
<b>Reason for Blocked:</b>	<b>Hardware Configuration:</b>
<b>Needs QA:</b> Yes	<b>ChangeLog Required:</b> No
<b>Needs Doc:</b> Yes	
<b>Description</b>	
<pre>[root@z30ref-a] ~# zilstat -p tank dtrace: invalid probe specifier  ok....  [root@z30ref-a] /usr/local# dtrace -s ./libexec/freenas-debug/dtrace/disklatencycmd.d dtrace: failed to compile script ./libexec/freenas-debug/dtrace/disklatencycmd.d: line 34: operator -&gt; cannot be applied to a forward declaration: no struct devstat definition is available  Something is unhappy</pre>	
<b>Related issues:</b>	
Has duplicate FreeNAS - Bug #7235: zilstat not working after 9.3 upgrade <span style="float: right;"><b>Closed: Duplicate #21/2014</b></span>	

### Associated revisions

#### Revision 569ec27c - 01/08/2015 09:38 AM - kmoore

- Add patch to fix providing the full pathname for kernel + modules  
This fixes using dtrace when booting from grub, and perhaps others  
that expect full pathname in kldstat(2)  
- Bump PORTREV

Ticket: #6521

git-svn-id: svn+ssh://svn.freebsd.org/ports/head@376537 35697150-7ecd-e111-bb59-0022644237b5  
(cherry picked from commit 5e165a0048d5bd0d36c9c3835290f9e48d6b525d)  
(cherry picked from commit 4626fba50b9532e4745f65cdf37074d143e29cc)

### History

#### #1 - 11/11/2014 06:19 AM - Linda Kateley

I am seeing this same error in a number of dtrace scripts in FreeBSD 10.0, the "invalid probe specifier".

#### #2 - 11/14/2014 08:30 PM - Josh Paetzel

- Status changed from Unscreened to Screened

**#3 - 11/23/2014 10:26 AM - Josh Paetzel**

- Status changed from Screened to Investigation

I had hopes for a506c75 but alas, that was not the solution.

**#4 - 11/23/2014 04:42 PM - Josh Paetzel**

- Status changed from Investigation to Unscreened

- Assignee changed from Josh Paetzel to Jordan Hubbard

```
root@freenas] /usr/local# dtrace -s ./libexec/freenas-debug/dtrace/disklatencycmd.d
dtrace: failed to compile script ./libexec/freenas-debug/dtrace/disklatencycmd.d: line 34: operator -> cannot be applied to a forward declaration: no
struct devstat definition is available
```

```
[root@freenas] /usr/local# zilstat -p tank
: in action list: index 1 is out of range for fbt::zil_lwb_write_start:entry args[ ]
```

I'm out of ideas as to what could be wrong.

**#5 - 11/24/2014 04:43 PM - Xin Li**

- Status changed from Unscreened to Screened

Revisions [trueos|a9d3a42](#) and [trueos|a853a81](#) may be related to this issue.

I have also integrated [trueos|c6b7a2b](#) and [trueos|4c31dea](#) while there.

**#6 - 11/26/2014 09:09 AM - Jordan Hubbard**

- Assignee changed from Jordan Hubbard to Xin Li

I have no ideas either. :(

Apparently DTrace is still broken even after the latest changes. How weird.

**#7 - 11/30/2014 12:57 PM - Jordan Hubbard**

- Priority changed from Nice to have to Important

**#8 - 12/01/2014 12:09 PM - Jordan Hubbard**

- Assignee changed from Xin Li to Josh Paetzel

**#9 - 12/02/2014 11:30 AM - Josh Paetzel**

- Status changed from Screened to Investigation

This has been narrowed down to a grub problem.

TrueOS works fine if booted from the FreeBSD bootloader.

**#10 - 12/02/2014 11:45 AM - Josh Paetzel**

- Status changed from Investigation to Closed: Behaves correctly

**#11 - 12/02/2014 11:48 AM - Josh Paetzel**

- Status changed from Closed: Behaves correctly to Investigation

**#12 - 12/04/2014 12:46 PM - Jordan Hubbard**

- Target version changed from 9.3-RELEASE to Unspecified

- Seen in changed from 9.3-M4 to 9.3-BETA

BRB: This is still a real bug that needs fixing, but we have determined that it is not a release blocker. Marking as SU Candidate.

**#13 - 12/11/2014 10:55 AM - Dimitar Boyn**

In **FreeNAS-9.3-STABLE-201412090314**

```
~# zilstat fastpool
dtrace: invalid probe specifier
#pragma D option quiet
  inline int OPT_time = 0;
  inline int OPT_txg = 0;
  inline int OPT_pool = 0;
  inline int OPT_mega = 0;
  inline int INTERVAL = fastpool;
  inline int LINES = -1;
  inline int COUNTER = -1;
  inline int FILTER = 0;
  inline string POOL = "";
dtrace:::BEGIN
{
  /* starting values */
  MEGA = 1000000;
  counts = COUNTER;
  secs = INTERVAL;
  interval = INTERVAL;
  interval == 0 ? interval++ : 1;
  line = 0;
  last_event[""] = 0;
  nused=0;
  nused_max_per_sec=0;
  nused_per_sec=0;
  size=0;
  size_max_per_sec=0;
  size_per_sec=0;
  syncops=0;
  size_4k=0;
  size_4k_32k=0;
  size_32k=0;
  OPT_txg ? printf("waiting for txg commit...\n") : 1;
}

/*
 * collect info when zil_lwb_write_start fires
 */
fbt::zil_lwb_write_start:entry
/OPT_pool == 0 || POOL == args[0]->z1_dmu_pool->dp_spa->spa_name/
{
  nused += args[1]->lwb_nused;
  nused_per_sec += args[1]->lwb_nused;
  size += args[1]->lwb_sz;
  size_per_sec += args[1]->lwb_sz;
  syncops++;
  args[1]->lwb_sz <= 4096 ? size_4k++ : 1;
  args[1]->lwb_sz > 4096 && args[1]->lwb_sz < 32768 ? size_4k_32k++ : 1;
  args[1]->lwb_sz >= 32768 ? size_32k++ : 1;
}

/*
```

```

* Timer
*/
profile:::tick-1sec
{
    OPT_txx ? secs++ : secs--;
    nused_per_sec > nused_max_per_sec ? nused_max_per_sec = nused_per_sec : 1;
    nused_per_sec = 0;
    size_per_sec > size_max_per_sec ? size_max_per_sec = size_per_sec : 1;
    size_per_sec = 0;
}

/*
* Print header
*/
profile:::tick-1sec
/OPT_txx == 0 && line == 0/
{
    /* print optional headers */
    OPT_time ? printf("%-20s ", "TIME") : 1;

    /* print header */
    OPT_mega ? printf("%10s %10s %10s %10s %10s %10s",
        "N-MB", "N-MB/s", "N-Max-Rate",
        "B-MB", "B-MB/s", "B-Max-Rate") :
        printf("%10s %10s %10s %10s %10s %10s",
            "N-Bytes", "N-Bytes/s", "N-Max-Rate",
            "B-Bytes", "B-Bytes/s", "B-Max-Rate");
    printf(" %6s %6s %6s %6s\n",
        "ops", "<=4kB", "4-32kB", ">=32kB");
    line = LINES;
}

fbt::txg_quiesce:entry
/OPT_txx == 1 && POOL == args[0]->dp_spa->spa_name && line == 0/
{
    OPT_time ? printf("%-20s ", "TIME") : 1;

    OPT_mega ? printf("%10s %10s %10s %10s %10s %10s %10s",
        "txg", "N-MB", "N-MB/s", "N-Max-Rate",
        "B-MB", "B-MB/s", "B-Max-Rate") :
        printf("%10s %10s %10s %10s %10s %10s %10s",
            "txg", "N-Bytes", "N-Bytes/s", "N-Max-Rate",
            "B-Bytes", "B-Bytes/s", "B-Max-Rate");
    printf(" %6s %6s %6s %6s\n",
        "ops", "<=4kB", "4-32kB", ">=32kB");
    line = LINES;
}

/*
* Print Output
*/
profile:::tick-1sec
/OPT_txx == 0 && secs == 0/
{
    OPT_time ? printf("%-20Y ", walltimestamp) : 1;
    OPT_mega ?
        printf("%10d %10d %10d %10d %10d %10d",
            nused/MEGA, nused/(interval*MEGA), nused_max_per_sec/MEGA,
            size/MEGA, size/(interval*MEGA), size_max_per_sec/MEGA) :
        printf("%10d %10d %10d %10d %10d %10d",
            nused, nused/interval, nused_max_per_sec,
            size, size/interval, size_max_per_sec);
    printf(" %6d %6d %6d %6d\n",
        syncops, size_4k, size_4k_32k, size_32k);
    nused = 0;
    nused_per_sec = 0;
    nused_max_per_sec = 0;
    size=0;
    size_max_per_sec=0;
    size_per_sec=0;
    syncops=0;
    size_4k=0;
    size_4k_32k=0;
    size_32k=0;
    secs = INTERVAL;
}

```

```

counts--;
line--;
}

fbt::txg_quiesce:entry
/OPT_txx == 1 && POOL == args[0]->dp_spa->spa_name/
{
secs <= 0 ? secs=1 : 1;
OPT_time ? printf("%-20Y ", walltimestamp) : 1;
OPT_mega ?
printf("%10d %10d %10d %10d %10d %10d %10d", args[1],
nused/MEGA, nused/(secs*MEGA), nused_max_per_sec/MEGA,
size/MEGA, size/(secs*MEGA), size_max_per_sec/MEGA) :
printf("%10d %10d %10d %10d %10d %10d %10d", args[1],
nused, nused/secs, nused_max_per_sec,
size, size/secs, size_max_per_sec);
printf(" %6d %6d %6d %6d\n",
syncops, size_4k, size_4k_32k, size_32k);
nused = 0;
nused_per_sec = 0;
nused_max_per_sec = 0;
size=0;
size_max_per_sec=0;
size_per_sec=0;
syncops=0;
size_4k=0;
size_4k_32k=0;
size_32k=0;
secs = 0;
counts--;
line--;
}

/*
* End of program
*/
profile::tick-1sec
/OPT_txx == 0 && counts == 0/
{
exit(0);
}
fbt::txg_quiesce:entry
/OPT_txx == 1 && counts == 0/
{
exit(0);
}
: "/usr/lib/dtrace/io.d", line 43: operator -> cannot be applied to a forward declaration: no struct devstat definition is available

```

**#14 - 12/16/2014 02:53 PM - Jordan Hubbard**

Yep, we know. We still have no idea how/where this broke.

**#15 - 12/21/2014 03:16 PM - Jordan Hubbard**

- Has duplicate Bug #7235: zilstat not working after 9.3 upgrade added

**#16 - 12/31/2014 10:08 AM - HonYin Kok**

Do we have an eta for fix to becomes available? I think the problem is dtrace related. Maybe because the kernel uses ithread instead of thread?

:"/usr/lib/dtrace/psinfo.d", line 89: failed to resolve type kernel`struct thread \* for identifier curthread: Module is no longer loaded

```
1. dtrace -l | grep thread
549 fbt kernel xpt_scanner_thread entry
857 fbt kernel ctl_work_thread entry
858 fbt kernel ctl_lun_thread entry
859 fbt kernel ctl_thresh_thread entry
1063 fbt kernel cfiscsi_maintenance_thread entry
1561 fbt kernel l2arc_feed_thread entry
1608 fbt kernel arc_reclaim_thread entry
1790 fbt kernel traverse_prefetch_thread entry
1791 fbt kernel traverse_prefetch_thread return
2198 fbt kernel spa_async_thread_vd entry
2199 fbt kernel spa_async_thread entry
2253 fbt kernel trim_thread entry
2256 fbt kernel txg_thread_enter entry
2257 fbt kernel txg_thread_enter return
2258 fbt kernel txg_thread_wait entry
2259 fbt kernel txg_thread_wait return
2260 fbt kernel txg_thread_exit entry
2262 fbt kernel txg_quiesce_thread entry
2263 fbt kernel txg_sync_thread entry
3463 fbt kernel db_trace_thread_wrapper entry
3464 fbt kernel db_trace_thread_wrapper return
3475 fbt kernel thread_show_add entry
3478 fbt kernel thread_show_del entry
3480 fbt kernel db_show_thread entry
3481 fbt kernel dumpthread entry
3482 fbt kernel dumpthread return
3576 fbt kernel aac_command_thread entry
3712 fbt kernel aac_command_thread entry
4211 fbt kernel acpi_tz_cooling_thread_start entry
4212 fbt kernel acpi_tz_cooling_thread_start return
4213 fbt kernel acpi_tz_cooling_thread entry
4234 fbt kernel acpi_tz_thread entry
4650 fbt kernel ahd_recovery_thread entry
4767 fbt kernel ahc_recovery_thread entry
5835 fbt kernel ciss_notify_thread entry
7370 fbt kernel fw_bus_probe_thread entry
7871 fbt kernel icl_send_thread entry
7872 fbt kernel icl_receive_thread entry
7906 fbt kernel iscsi_maintenance_thread entry
7964 fbt kernel isp_kthread entry
8615 fbt kernel md_kthread entry
9531 fbt kernel mpt_recovery_thread entry
9599 fbt kernel mpt_raid_thread entry
10276 fbt kernel random_kthread entry
10277 fbt kernel random_kthread return
15081 fbt kernel elf64_note_threadmd entry
15082 fbt kernel elf64_note_threadmd return
15128 fbt kernel acct_thread entry
15313 fbt kernel kdtrace_thread_ctor entry
15314 fbt kernel kdtrace_thread_ctor return
15315 fbt kernel kdtrace_thread_dtor entry
15316 fbt kernel kdtrace_thread_dtor return
15444 fbt kernel intr_event_schedule_thread entry
15445 fbt kernel intr_event_schedule_thread return
15448 fbt kernel ithread_update entry
15451 fbt kernel ithread_loop entry
15745 fbt kernel fill_kinfo_thread entry
15746 fbt kernel fill_kinfo_thread return
15865 fbt kernel sig_suspend_threads entry
```

15866	fbt	kernel	sig_suspend_threads return
15945	fbt	kernel	create_thread entry
15946	fbt	kernel	create_thread return
15949	fbt	kernel	thread_fini entry
15950	fbt	kernel	thread_init entry
15951	fbt	kernel	thread_init return
15952	fbt	kernel	thread_dtor entry
15953	fbt	kernel	thread_dtor return
15954	fbt	kernel	thread_ctor entry
15955	fbt	kernel	thread_ctor return
15985	fbt	kernel	get_thread_cputime entry
15986	fbt	kernel	get_thread_cputime return
16009	fbt	kernel	umtx_pi_adjust_thread entry
16010	fbt	kernel	umtx_pi_adjust_thread return
16015	fbt	kernel	umtx_thread_cleanup entry
16016	fbt	kernel	umtx_thread_cleanup return
16156	fbt	kernel	sched_thread_priority entry
16157	fbt	kernel	sched_thread_priority return
16426	fbt	kernel	sleepq_resume_thread entry
16427	fbt	kernel	sleepq_resume_thread return
16471	fbt	kernel	taskqueue_define_thread entry
16489	fbt	kernel	print_thread entry
16509	fbt	kernel	turnstile_adjust_thread entry
16510	fbt	kernel	turnstile_adjust_thread return
18304	fbt	kernel	ngthread entry
19627	fbt	kernel	xprt_assignthread entry
19628	fbt	kernel	xprt_assignthread return
19631	fbt	kernel	svc_new_thread entry
19632	fbt	kernel	svc_new_thread return
19633	fbt	kernel	svc_thread_start entry
19640	fbt	kernel	svcpool_minthread_sysctl entry
19641	fbt	kernel	svcpool_minthread_sysctl return
19642	fbt	kernel	svcpool_maxthread_sysctl entry
19643	fbt	kernel	svcpool_maxthread_sysctl return
19644	fbt	kernel	svcpool_threads_sysctl entry
19645	fbt	kernel	svcpool_threads_sysctl return
20506	fbt	kernel	vm_thread_stack_lowmem entry
20507	fbt	kernel	vm_thread_stack_lowmem return
20508	fbt	kernel	vm_thread_stack_dispose entry
23476	fbt	kernel	fdc_thread entry
23574	fbt	kernel	hpt_worker_thread entry
23598	fbt	kernel	thread_io_done entry
25450	fbt	kernel	vtballoon_thread entry
25664	fbt	kernel	elf32_note_threadmd entry
25665	fbt	kernel	elf32_note_threadmd return
26582	fbt	kernel	thread_unthread entry
26583	fbt	kernel	thread_unthread return
27064	fbt	kernel	trim_thread_create entry
27065	fbt	kernel	trim_thread_create return
27621	fbt	kernel	kthread_add entry
27622	fbt	kernel	kthread_add return
27814	fbt	kernel	dev_relthread entry
27815	fbt	kernel	dev_relthread return
27994	fbt	kernel	vm_thread_dispose entry
27995	fbt	kernel	vm_thread_dispose return
28675	fbt	kernel	threadinit entry
28781	fbt	kernel	nfsccl_renewthread entry
28782	fbt	kernel	nfsccl_renewthread return
29328	fbt	kernel	db_set_thread entry
29329	fbt	kernel	db_set_thread return
29506	fbt	kernel	trim_thread_destroy entry
29507	fbt	kernel	trim_thread_destroy return
30122	fbt	kernel	thread_free entry
30146	fbt	kernel	kdtrace_thread_size entry
30147	fbt	kernel	kdtrace_thread_size return
30730	fbt	kernel	cpuset_setthread entry
30731	fbt	kernel	cpuset_setthread return
31220	fbt	kernel	thread_unlink entry
31221	fbt	kernel	thread_unlink return
31457	fbt	kernel	sched_fork_thread entry
31458	fbt	kernel	sched_fork_thread return
32297	fbt	kernel	ahc_spawn_recovery_thread entry
32298	fbt	kernel	ahc_spawn_recovery_thread return
32398	fbt	kernel	sys_rtprio_thread entry
32399	fbt	kernel	sys_rtprio_thread return

32663	fbt	kernel	thread_single_end entry
32664	fbt	kernel	thread_single_end return
33561	fbt	kernel	thread_suspend_switch entry
33562	fbt	kernel	thread_suspend_switch return
33742	fbt	kernel	taskqueue_thread_loop entry
34089	fbt	kernel	umtx_thread_alloc entry
34090	fbt	kernel	umtx_thread_alloc return
34184	fbt	kernel	cpu_thread_exit entry
34185	fbt	kernel	cpu_thread_exit return
34560	fbt	kernel	taskqueue_thread_enqueue entry
35559	fbt	kernel	thread_stash entry
36721	fbt	kernel	mac_thread_userret entry
36722	fbt	kernel	mac_thread_userret return
37057	fbt	kernel	devvn_refthread entry
37058	fbt	kernel	devvn_refthread return
37448	fbt	kernel	thread_lock_block entry
37449	fbt	kernel	thread_lock_block return
37502	fbt	kernel	thread_alloc entry
37503	fbt	kernel	thread_alloc return
37766	fbt	kernel	kdb_backtrace_thread entry
37767	fbt	kernel	kdb_backtrace_thread return
38149	fbt	kernel	cpu_thread_swapout entry
38150	fbt	kernel	cpu_thread_swapout return
38806	fbt	kernel	thread_link entry
38807	fbt	kernel	thread_link return
39242	fbt	kernel	db_print_thread entry
39312	fbt	kernel	taskqueue_start_threads entry
39313	fbt	kernel	taskqueue_start_threads return
39316	fbt	kernel	trim_thread_wakeup entry
39317	fbt	kernel	trim_thread_wakeup return
39385	fbt	kernel	thread_unsuspend_one entry
39525	fbt	kernel	nfsc_start_renewthread entry
40110	fbt	kernel	syscall_thread_enter entry
40111	fbt	kernel	syscall_thread_enter return
40263	fbt	kernel	is_fpu_kern_thread entry
40264	fbt	kernel	is_fpu_kern_thread return
40714	fbt	kernel	sched_exit_thread entry
41182	fbt	kernel	cred_update_thread entry
41183	fbt	kernel	cred_update_thread return
41327	fbt	kernel	cpu_thread_alloc entry
41328	fbt	kernel	cpu_thread_alloc return
41374	fbt	kernel	thread_reap entry
41375	fbt	kernel	thread_reap return
41477	fbt	kernel	thread_lock_unblock entry
41478	fbt	kernel	thread_lock_unblock return
41728	fbt	kernel	thread_unsuspend entry
41729	fbt	kernel	thread_unsuspend return
42241	fbt	kernel	kthread_suspend entry
42242	fbt	kernel	kthread_suspend return
43335	fbt	kernel	dev_refthread entry
43336	fbt	kernel	dev_refthread return
43588	fbt	kernel	thread_stopped entry
43589	fbt	kernel	thread_stopped return
43600	fbt	kernel	cpu_thread_free entry
44204	fbt	kernel	thread_single entry
44205	fbt	kernel	thread_single return
44340	fbt	kernel	kthread_resume entry
44341	fbt	kernel	kthread_resume return
44398	fbt	kernel	db_lookup_thread entry
44399	fbt	kernel	db_lookup_thread return
44546	fbt	kernel	sched_sizeof_thread entry
44547	fbt	kernel	sched_sizeof_thread return
44602	fbt	kernel	audit_thread_alloc entry
44603	fbt	kernel	audit_thread_alloc return
45008	fbt	kernel	thread_suspend_one entry
45296	fbt	kernel	elf32_dump_thread entry
45297	fbt	kernel	elf32_dump_thread return
45367	fbt	kernel	thread_wait entry
45385	fbt	kernel	kproc_kthread_add entry
45386	fbt	kernel	kproc_kthread_add return
45758	fbt	kernel	thread_lock_set entry
45828	fbt	kernel	ahc_terminate_recovery_thread entry
45829	fbt	kernel	ahc_terminate_recovery_thread return
45834	fbt	kernel	db_trace_thread entry
45999	fbt	kernel	ahd_spawn_recovery_thread entry



46000	fbt	kernel	ahd_spawn_recovery_thread return
46383	fbt	kernel	_thread_lock_flags entry
46384	fbt	kernel	_thread_lock_flags return
46653	fbt	kernel	syscall_thread_exit entry
46654	fbt	kernel	syscall_thread_exit return
46840	fbt	kernel	kthread_shutdown entry
46841	fbt	kernel	kthread_shutdown return
47130	fbt	kernel	ahd_terminate_recovery_thread entry
47131	fbt	kernel	ahd_terminate_recovery_thread return
47557	fbt	kernel	db_show_threads entry
47558	fbt	kernel	db_show_threads return
47642	fbt	kernel	kthread_start entry
47643	fbt	kernel	kthread_start return
47652	fbt	kernel	thread_find entry
47653	fbt	kernel	thread_find return
47753	fbt	kernel	thread_alloc_stack entry
47754	fbt	kernel	thread_alloc_stack return
49259	fbt	kernel	umtx_thread_init entry
49260	fbt	kernel	umtx_thread_init return
49281	fbt	kernel	cpu_thread_swapin entry
49282	fbt	kernel	cpu_thread_swapin return
49394	fbt	kernel	fpu_kern_thread entry
49395	fbt	kernel	fpu_kern_thread return
49778	fbt	kernel	kthread_exit entry
49779	fbt	kernel	callb_add_thread entry
49961	fbt	kernel	elf64_dump_thread entry
49962	fbt	kernel	elf64_dump_thread return
50020	fbt	kernel	cpu_thread_clean entry
50021	fbt	kernel	cpu_thread_clean return
50550	fbt	kernel	kthread_suspend_check entry
50551	fbt	kernel	kthread_suspend_check return
50728	fbt	kernel	audit_thread_free entry
50729	fbt	kernel	audit_thread_free return
51051	fbt	kernel	choosethread entry
51052	fbt	kernel	choosethread return
51081	fbt	kernel	vm_thread_new entry
51082	fbt	kernel	vm_thread_new return
51579	fbt	kernel	thread_suspend_check entry
51580	fbt	kernel	thread_suspend_check return
51591	fbt	kernel	thread_exit entry
51622	fbt	kernel	umtx_thread_fini entry
51692	fbt	kernel	umtx_thread_exit entry
52168	fbt	kernel	cpuset_thread0 entry
52169	fbt	kernel	cpuset_thread0 return
55222	dtmalloc		ithread malloc
55223	dtmalloc		ithread free
55460	lockstat	kernel	thread_lock thread-spin
56375	syscall	freebsd	rtprio_thread entry
56376	syscall	freebsd	rtprio_thread return
57447	syscall	freebsd32	rtprio_thread entry
57448	syscall	freebsd32	rtprio_thread return

#17 - 01/06/2015 02:49 PM - titan\_rw -

Hi All.

I too am missing the functionality of zilstat.

I had a possibly stupid idea.

I understand this broke since moving to having the system boot with grub. Could this be due to having two different zpool.cache files? I noticed this trying to get "zdb -Dv tank" to work:

```
root@nas2 ~ # zdb -Dv nas2pool
zdb: can't open 'nas2pool': No such file or directory
root@nas2 ~ # zpool status nas2pool
pool: nas2pool
state: ONLINE
scan: resilvered 1.20T in 6h43m with 0 errors on Tue Dec 30 21:04:05 2014
—snip—
root@nas2 ~ # zdb -C
freenas-boot:
version: 28
name: 'freenas-boot'
state: 0
txg: 1046770
pool_guid: 2093658414403048516
hostid: 3187196706
```

(only freenas-boot listed)

```
root@nas2 ~ # zdb -U /data/zfs/zpool.cache
nas2pool:
version: 5000
name: 'nas2pool'
state: 0
txg: 563273
pool_guid: 3017840465254555211
hostid: 3187196706
root@nas2 ~ # zdb -D -U /data/zfs/zpool.cache nas2pool
DDT-sha256-zap-duplicate: 2481952 entries, size 1115 on disk, 180 in core
DDT-sha256-zap-unique: 71143 entries, size 34247 on disk, 5534 in core

dedup = 16.95, compress = 1.08, copies = 1.00, dedup * compress / copies = 18.28
```

It wouldn't be that zilstat, or dtrace is somehow referencing /boot/zfs/zpool.cache or something?

I assume not booting off of grub would result in not having the two cache files?

Just a shot in the dark anyway.

**#18 - 01/06/2015 04:12 PM - Jordan Hubbard**

Thanks for the suggestion. Sadly, dtrace has absolutely no knowledge of ZFS (or ZFS cache files).

**#19 - 01/07/2015 03:31 PM - Josh Paetzel**

I have faith that Kip Macy will be able to fix this.

**#20 - 01/08/2015 02:07 AM - Jordan Hubbard**

- Status changed from Investigation to Fix In Progress

See the nightly.. The sample script now prints:

```
[root@freenas-test] ~# dtrace -s /usr/local/libexec/freenas-debug/dtrace/disklatencycmd.d
dtrace: buffer size lowered to 8m
^C
total time, us: 10000702
ada (1), us:
```

value	Distribution	count
< 1000	@oooooooooooooooooooooooooooo	141
1000		0
...		

I couldn't sleep, and this problem was bothering me. :-/

**#21 - 01/08/2015 09:40 AM - Josh Paetzel**

- Status changed from Fix In Progress to Ready For Release

- Seen in changed from 9.3-BETA to 9.3-RELEASE

Thanks everyone for their help. As of this morning the fix has been confirmed.

**#22 - 01/08/2015 12:53 PM - Jordan Hubbard**

- Status changed from Ready For Release to Fix In Progress

- Assignee changed from Josh Paetzel to Sean Fagan

BRB: The fix is in for new installs, but we need to deal with re-installing grub (the fixed grub) for existing ones on upgrade. If sef can get this done in the next day, we'll do that, otherwise I'll bring back my hack-fix which works-around the problem.

**#23 - 01/08/2015 01:02 PM - titan\_rw -**

Awesome.

Any ideas on when this fix will make it into -BETA, or even -STABLE?

Kudos to whoever figured it out. Not long you see something that (to me at least) appeared to defy reason.

**#24 - 01/08/2015 04:03 PM - Sean Fagan**

- Assignee changed from Sean Fagan to Xin Li

We're not going to be able to make the next update with this -- it's going to take more than a day.

I ran into two issues, one very annoying and one a blocker:

- 1) "chroot /mnt sh -c 'mount -t devfs devfs /dev ; umount -f /dev' will fail. I think devfs is looking at the wrong path when it goes to unmount. I can live with this one (it just leaves things mounted, but we're rebooting anyway).
- 2) grub-install fails of /boot/grub is a nullfs mount. I am doing "grub-install '--modules=zfs part\_gpt' /dev/da0", and it is saying Installing for i386-pc platform.  
grub-install: error: cannot find a device for /boot/grub (is /dev mounted?).

If I unmount \${mount}/boot/grub from outside the chroot first, it works. The problem is that the installer code that's already there sets up the mount points, and I believe it should continue to do so.

Back to Xin for this. We can discuss; we may want to make another bug for the devfs issue.

Note that what I was working on was to add a post-update script to base-os:

```
for disk in $(sysctl -n kern.disks); do
    mounted=false
    if [ ! -c /dev/${disk} ]; then
        mount -t devfs devfs /dev && mounted=true
    fi
    if gpart show ${disk} | grep -q bios-boot; then
        /usr/local/sbin/grub-install --modules='zfs part_gpt' /dev/${disk}
    fi
    if ${mounted} ; then umount -f /dev ; fi
done
```

I am really not too happy with that -- it'll install on *any* disk in the system that has a bios-boot partition -- but the alternative is a way to get the physical disks for a pool, and I'm not sure how to do that. I also don't like that this means grub would be re-installed on every update that includes base-os; we need a way to know what version of grub is installed, and what the new version is.

**#25 - 01/08/2015 05:40 PM - Jordan Hubbard**

- Copied to Bug #7438: Dtrace is unhappy added

**#26 - 01/08/2015 05:41 PM - Jordan Hubbard**

- Status changed from Fix In Progress to Ready For Release

Copied to [#7438](#) to track "the right fix" but for now, this can be marked Resolved / RFR since my hack-fix deals with the issue well enough to close the original bug.

**#27 - 01/08/2015 05:41 PM - Jordan Hubbard**

- Assignee changed from Xin Li to Jordan Hubbard

**#28 - 01/08/2015 06:09 PM - Jordan Hubbard**

- Status changed from Ready For Release to Resolved

**#29 - 01/08/2015 08:06 PM - Josh Paetzel**

- Status changed from Resolved to Unscreened

The hacky fix does not get zilstat working, however it does get disklatency.d working.

**#30 - 01/08/2015 09:14 PM - Jordan Hubbard**

- Status changed from Unscreened to Resolved

Back to Resolved - this is not the right bug to go forward with, since the fix it's tracking is already in. The bug for the continuing investigation is [#7438](#), as above.

**#31 - 01/15/2015 12:14 PM - Jordan Hubbard**

- Copied to deleted (Bug #7438: Dtrace is unhappy)

**#32 - 08/26/2016 01:53 PM - Kris Moore**

- Target version changed from Unspecified to N/A