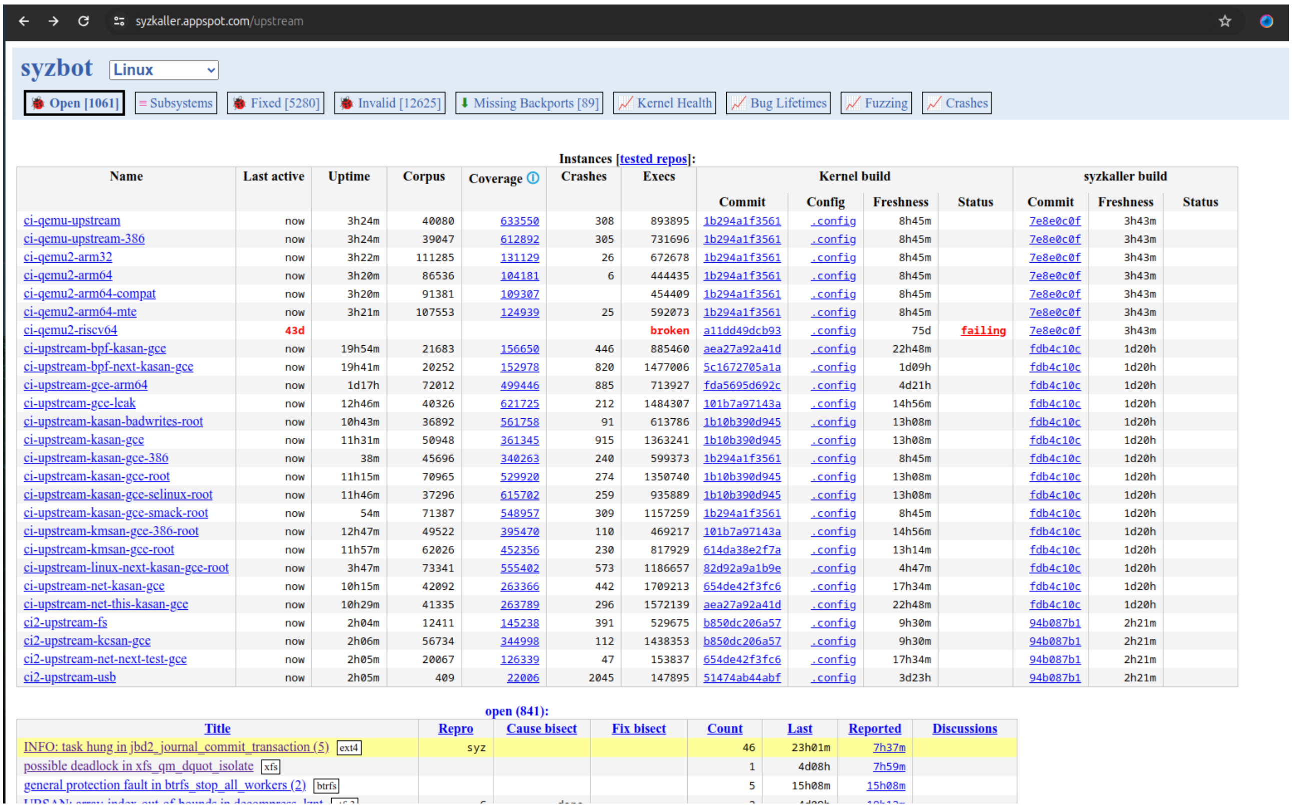
LKMP stacktrace

The decode_stacktrace script (https://github.com/torvalds/linux/blob/master/scripts/decode_stacktrace.sh) decodes the stack dump, translating all kernel addresses in the stack dump into line numbers. Therefore making the stack dump friendlier to work with.

- reference: https://lwn.net/Articles/592724/

The last bug found in syzbot is: https://syzkaller.appspot.com/bug?extid=3071bdd0a9953bc0d177



Instances [tested repos]:

Name	Last active	Uptime	Corpus	Coverage ⓘ	Crashes	Execs	Kernel build				syzkaller build		
							Commit	Config	Freshness	Status	Commit	Freshness	Status
ci-qemu-upstream	now	3h24m	40080	633550	308	893895	1b294a1f3561	.config	8h45m		7e8e0c0f	3h43m	
ci-qemu-upstream-386	now	3h24m	39047	612892	305	731696	1b294a1f3561	.config	8h45m		7e8e0c0f	3h43m	
ci-qemu2-arm32	now	3h22m	111285	131129	26	672678	1b294a1f3561	.config	8h45m		7e8e0c0f	3h43m	
ci-qemu2-arm64	now	3h20m	86536	104181	6	444435	1b294a1f3561	.config	8h45m		7e8e0c0f	3h43m	
ci-qemu2-arm64-compat	now	3h20m	91381	109307		454409	1b294a1f3561	.config	8h45m		7e8e0c0f	3h43m	
ci-qemu2-arm64-mte	now	3h21m	107553	124939	25	592073	1b294a1f3561	.config	8h45m		7e8e0c0f	3h43m	
ci-qemu2-riscv64	43d					broken	a11dd49dcb93	.config	75d	failing	7e8e0c0f	3h43m	
ci-upstream-bpf-kasan-gce	now	19h54m	21683	156650	446	885460	aea27a92a41d	.config	22h48m		fdb4c10c	1d20h	
ci-upstream-bpf-next-kasan-gce	now	19h41m	20252	152978	820	1477006	5c1672705a1a	.config	1d09h		fdb4c10c	1d20h	
ci-upstream-gce-arm64	now	1d17h	72012	499446	885	713927	fda5695d692c	.config	4d21h		fdb4c10c	1d20h	
ci-upstream-gce-leak	now	12h46m	40326	621725	212	1484307	101b7a97143a	.config	14h56m		fdb4c10c	1d20h	
ci-upstream-kasan-badwrites-root	now	10h43m	36892	561758	91	613786	1b10b390d945	.config	13h08m		fdb4c10c	1d20h	
ci-upstream-kasan-gce	now	11h31m	50948	361345	915	1363241	1b10b390d945	.config	13h08m		fdb4c10c	1d20h	
ci-upstream-kasan-gce-386	now	38m	45696	340263	240	599373	1b294a1f3561	.config	8h45m		fdb4c10c	1d20h	
ci-upstream-kasan-gce-root	now	11h15m	70965	529920	274	1350740	1b10b390d945	.config	13h08m		fdb4c10c	1d20h	
ci-upstream-kasan-gce-selinux-root	now	11h46m	37296	615702	259	935889	1b10b390d945	.config	13h08m		fdb4c10c	1d20h	
ci-upstream-kasan-gce-smack-root	now	54m	71387	548957	309	1157259	1b294a1f3561	.config	8h45m		fdb4c10c	1d20h	
ci-upstream-kmsan-gce-386-root	now	12h47m	49522	395470	110	469217	101b7a97143a	.config	14h56m		fdb4c10c	1d20h	
ci-upstream-kmsan-gce-root	now	11h57m	62026	452356	230	817929	614da38e2f7a	.config	13h14m		fdb4c10c	1d20h	
ci-upstream-linux-next-kasan-gce-root	now	3h47m	73341	555402	573	1186657	82d92a9a1b9e	.config	4h47m		fdb4c10c	1d20h	
ci-upstream-net-kasan-gce	now	10h15m	42092	263366	442	1709213	654de42f3fc6	.config	17h34m		fdb4c10c	1d20h	
ci-upstream-net-this-kasan-gce	now	10h29m	41335	263789	296	1572139	aea27a92a41d	.config	22h48m		fdb4c10c	1d20h	
ci2-upstream-fs	now	2h04m	12411	145238	391	529675	b850dc206a57	.config	9h30m		94b087b1	2h21m	
ci2-upstream-kcsan-gce	now	2h06m	56734	344998	112	1438353	b850dc206a57	.config	9h30m		94b087b1	2h21m	
ci2-upstream-net-next-test-gce	now	2h05m	20067	126339	47	153837	654de42f3fc6	.config	17h34m		94b087b1	2h21m	
ci2-upstream-usb	now	2h05m	409	22006	2045	147895	51474ab44abf	.config	3d23h		94b087b1	2h21m	

open (841):

Title	Repro	Cause bisect	Fix bisect	Count	Last	Reported	Discussions
INFO: task hung in jbd2_journal_commit_transaction (5) ext4	syz			46	23h01m	7h37m	
possible deadlock in xfs_qm_dquot_isolate xfs				1	4d08h	7h59m	
general protection fault in btrfs_stop_all_workers (2) btrfs				5	15h08m	15h08m	

Its report is here: https://syzkaller.appspot.com/text?tag=CrashReport&x=1371e8cc980000

```
INFO: task jbd2/sda1-8:4509 blocked for more than 143 seconds.
      Not tainted 6.9.0-next-20240513-syzkaller #0
"echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message.
task:jbd2/sda1-8     state:D
 stack:25008 pid:4509  tgid:4509  ppid:2      flags:0x00004000
Call Trace:
 <TASK>
 context_switch kernel/sched/core.c:5408 [inline]
 __schedule+0x17e8/0x4a50 kernel/sched/core.c:6745
 __schedule_loop kernel/sched/core.c:6822 [inline]
 schedule+0x14b/0x320 kernel/sched/core.c:6837
 io_schedule+0x8d/0x110 kernel/sched/core.c:9043
 bit_wait_io+0x12/0xd0 kernel/sched/wait_bit.c:209
 __wait_on_bit+0xb0/0x2f0 kernel/sched/wait_bit.c:49
```

```
out_of_line_wait_on_bit+0x1d5/0x260 kernel/sched/wait_bit.c:64
wait_on_buffer include/linux/buffer_head.h:415 [inline]
journal_wait_on_commit_record fs/jbd2/commit.c:171 [inline]
jbd2_journal_commit_transaction+0x3d7f/0x6760 fs/jbd2/commit.c:887
kjournald2+0x463/0x850 fs/jbd2/journal.c:201
kthread+0x2f0/0x390 kernel/kthread.c:389
ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147
ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244
</TASK>
```

The script ‘decode_stacktrace’, has been already applied there, as we can se that the kernel addresses have been transated into code lines.

Nevertheless, to apply the scritp:

```
        ./decode_stacktrace.sh [vmlinux] [base path]

Where vmlinux is the vmlinux to extract line numbers from and base path is
the path that points to the root of the build tree, for example:

        ./decode_stacktrace.sh vmlinux /home/sasha/linux/ < input.log >
output.log

The stack trace should be piped through it (I, for example, just pipe
the output of the serial console of my KVM test box through it).
```

This issue, seems to be lock related, where maybe the task hung because of a death-lock. I would try to resolve this issue, with the tool ‘lockdep’.