Richard WM Jones

JUNE 18, 2023 · 10:35 AM

Follow up to “I booted Linux 292,612 times”

Well that blew up. It was supposed to be just a silly off-the-cuff comment about how some bugs are very tedious to bisect.

To answer a few questions people had, here’s what actually happened. As they say, don’t believe everything you read in the press.

A few weeks ago I noticed that some nbdkit tests which work by booting a Linux appliance under qemu were randomly hanging. I ignored it to start off with, but it got annoying so I decided to try to track down what was going on. Initially we thought it might be a qemu bug so I started by filing a bug there and writing my thoughts as I went to investigate. After swapping qemu, Linux guest and Linux host versions around it became clear that the problem was probably in the Linux guest kernel (although I didn’t rule out an issue with KVM emulation which might have implicated either qemu or the host kvm.ko module).

Initially I just had a hang, and because getting to that hang involved booting Linux hundreds or thousands of times it wasn’t feasible to attach gdb at the start to trace through the hang. Instead I had to connect gdb after observing the hang. It turns out that when the Linux guest was “hanging” it really was just missing a timer event so the kernel was still running albeit making no progress. But the upshot is that the stack trace you see is not of the hang itself, but of an idle, slightly confused kernel. gdb was out of the picture.

But since guest kernel 6.0 seemed to work and 6.4rc seemed to hang, I had a path to bisecting the bug.

Well, a *very slow* path. You see there are 52,363 commits between those two kernels, which means at least 15 or 16 bisect steps. Each step was going to involve booting the kernel at least thousands times to prove it was working (if it hung before then I’d observe that).

I made the mistake here of not first working on a good test, instead just running “while guestfish … ; echo -n . ; done” and watching until I’d seen a page of dots to judge the kernel “good”. Yeah, that didn’t work. It turns out the hang was made more likely by slightly loading the test machine (or running the tests in parallel which is the same thing). As a result my first bisection that took several days got the wrong commit.

Back to the drawing board. This time I wrote a proper test. It booted the kernel 10,000 times using 8 threads, and checked the qemu output to see if the boot had hung, stop the test and print a diagnostic, or print “test ok” if it got through all iterations. This time my bisection was better but that still took a couple of days.

At that point I thought I had the right commit, but Paolo Bonzini suggested to me that I boot the kernel in parallel, in a loop, for 24 hours at the point immediately before the commit, to try to show that there was no latent issue in the kernel before. (As it turns out while this is a good idea, this analysis is subtly flawed as we’ll see).

So I did just that. After 21 hours I got bored (plus this is using a lot of electricity and generating huge amounts of heat, and we’re in the middle of a heatwave here in the UK). I killed the test after 292,612 successful boots.

I had a commit that looked suspicious, but what to do now? I posted my findings on LKML.

We still didn’t fully understand how to trigger the hang, except it was annoying and rare, seemed to happen with different frequencies on AMD and Intel, could be reproduced by several independent people, but crucially kernel developer Peter Zijlstra could not reproduce it.

[For the record, the bug is a load and hardware-speed dependent race condition. It will particularly affect qemu virtual machines, but at least in theory it

A commenter on Hacker News pointed out that simply inserting a sleep into the problematic code path caused the same hang (and I verified that). So the commit I had bisected to was the wrong one again – it exposed a latent bug simply because it ran the same code as a sleep. It was introducing the sleep which exposed the bug, not the commit I'd spent a week bisecting. And the 262K boots didn't in fact prove there was no latent bug. You live and learn …

Eventually the Amazon thread led to Thomas Gleixner suggesting a fix.

I tested the fix and … it worked!

Unfortunately the patch that introduced the bug has already gone into several stable trees meaning that many more people will likely be hitting the problem in future, but thanks to a heroic effort *of many people* (and not me, really) the bug has been fixed now.