Linux BPF Superpowers

05 Mar 2016

Last month I spoke at Facebook's Performance @Scale event about Linux BPF Superpowers. These are coming to Linux in the 4.x series, and I've been using them in new open source performance tools.

Video is on Facebook (30 mins):

Slides are on slideshare (PDF):

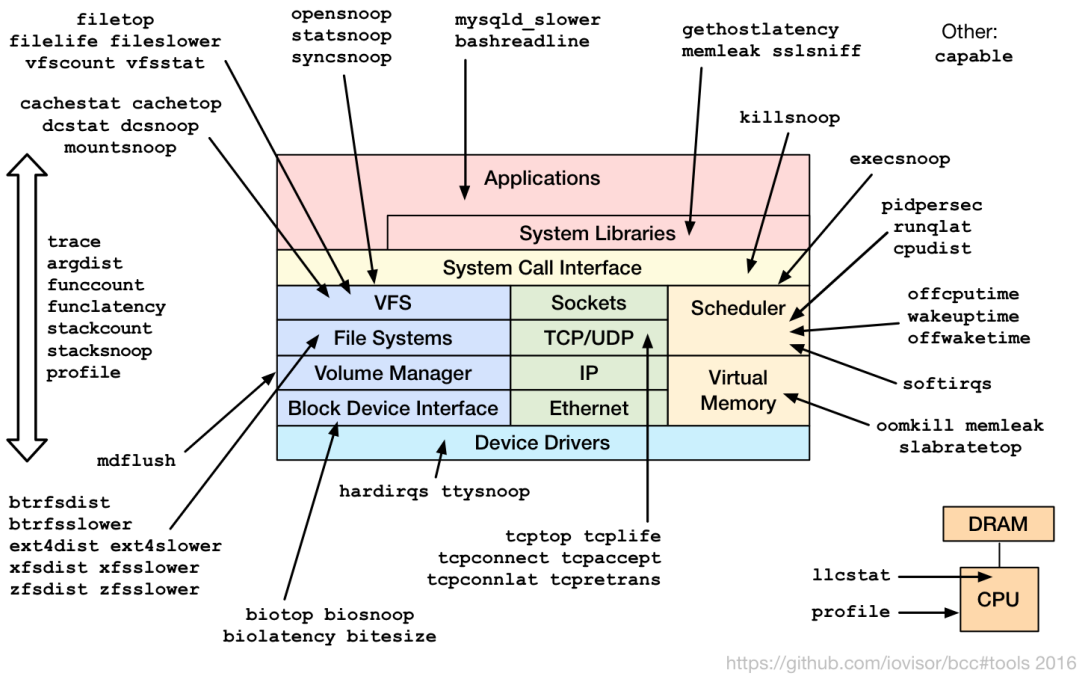
We've stopped calling it eBPF (extended Berkeley Packet Filter), and are now just calling it BPF, although we need a better backronym: Bytecode Probe Framework? Naming things is hard.

BPF is the in-kernel bytecode machine that can be used for tracing, virtual networks, and more. Alexei Starovoitov is the lead developer (he's now at Facebook), and there are developers from several companies contributing, including myself at Netflix, Daniel Borkmann at Cisco, and Brenden Blanco at PLUMgrid.

As an example of BPF, I opened with off-CPU analysis, and why BPF was making new things possible. I summarized some other examples as well, including gethostlatency, which instruments DNS lookups system wide without needing to restart anything:

```
# ./gethostlatency
TIME      PID    COMM           LATms HOST
06:10:24  28011  wget           90.00 www.iovisor.org
06:10:28  28127  wget            0.00 www.iovisor.org
06:10:41  28404  wget            9.00 www.netflix.com
06:10:48  28544  curl           35.00 www.netflix.com.au
06:11:10  29054  curl           31.00 www.plumgrid.com
06:11:16  29195  curl            3.00 www.facebook.com
06:11:25  29404  curl           72.00 foo
```

gethostlatency, and the other tools I demonstrated, are in bcc tools, which is a Python front end for BPF. For this talk I created a diagram of all the bcc tracing tools so far:



So many of my favourites (from other tracing languages) now have equivalents in bcc, which is pretty exciting. Tools like execsnoop, opensnoop, ext4slower, tcpretrans, tcpconnect, and runqlat.

These bcc tools are still in development and require at least Linux 4.1, which many people aren't running yet. You can think of them as a preview of things to come. But they are coming sooner rather than later: Ubuntu 16.04 (for example) will have a 4 series kernel, and isn't far away.

Please watch my talk video above, and check out the other talk videos which were pretty interesting as well (although that link plays the low-res versions in Chrome; high-res versions, like I linked to above, do exist).

Thanks to Facebook for having me – it was a great event.

Links from the talk

iovisor bcc:

- https://github.com/iovisor/bcc
- http://www.brendangregg.com/blog/2015-09-22/bcc-linux-4.3-tracing.html
- http://blogs.microsoft.co.il/sasha/2016/02/14/two-new-ebpf-tools-memleak-and-argdist/

BPF Off-CPU, Wakeup, Off-Wake & Chain Graphs:

- http://www.brendangregg.com/blog/2016-01-20/ebpf-offcpu-flame-graph.html
- http://www.brendangregg.com/blog/2016-02-01/linux-wakeup-offwake-profiling.html
- http://www.brendangregg.com/blog/2016-02-05/ebpf-chaingraph-prototype.html

Linux Performance:

- http://www.brendangregg.com/linuxperf.html

Linux perf_events:

- https://perf.wiki.kernel.org/index.php/Main_Page
- http://www.brendangregg.com/perf.html

Flame Graphs:

- http://techblog.netflix.com/2015/07/java-in-flames.html

- http://www.brendangregg.com/flamegraphs.html

Netflix Tech Blog on Vector:

- http://techblog.netflix.com/2015/04/introducing-vector-netflixs-on-host.html

---

You can comment here, but I can't guarantee your comment will remain here forever: I might switch comment systems at some point (eg, if disqus add advertisements).