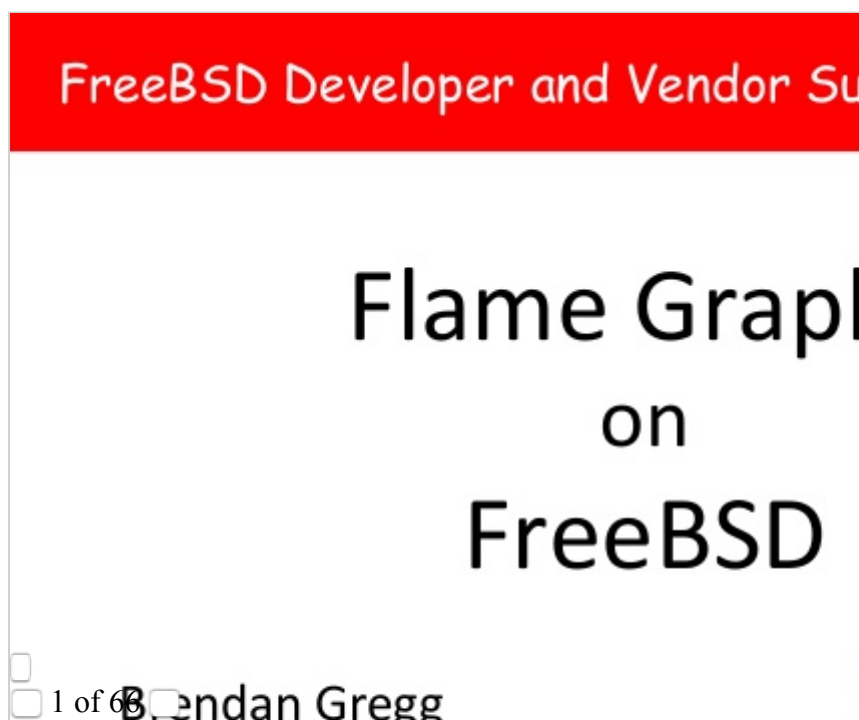


FreeBSD Flame Graphs

10 Mar 2015

At the last FreeBSD Developer and Vendor Summit, I gave a talk on "Flame Graphs for FreeBSD", where I summarized the different types (CPU, memory, disk I/O, off-CPU, chain graphs), showed how they can be generated on FreeBSD, and did some live demos. I think it's one of my best talks so far, whether you care about FreeBSD or not, to see how this visualization can be used to navigate different types of profiling data.

The slides are on [slideshare](#):



There was no camera in the room, so I captured the talk from my laptop using screenflow [youtube](#):

Flame Graphs on FreeBSD



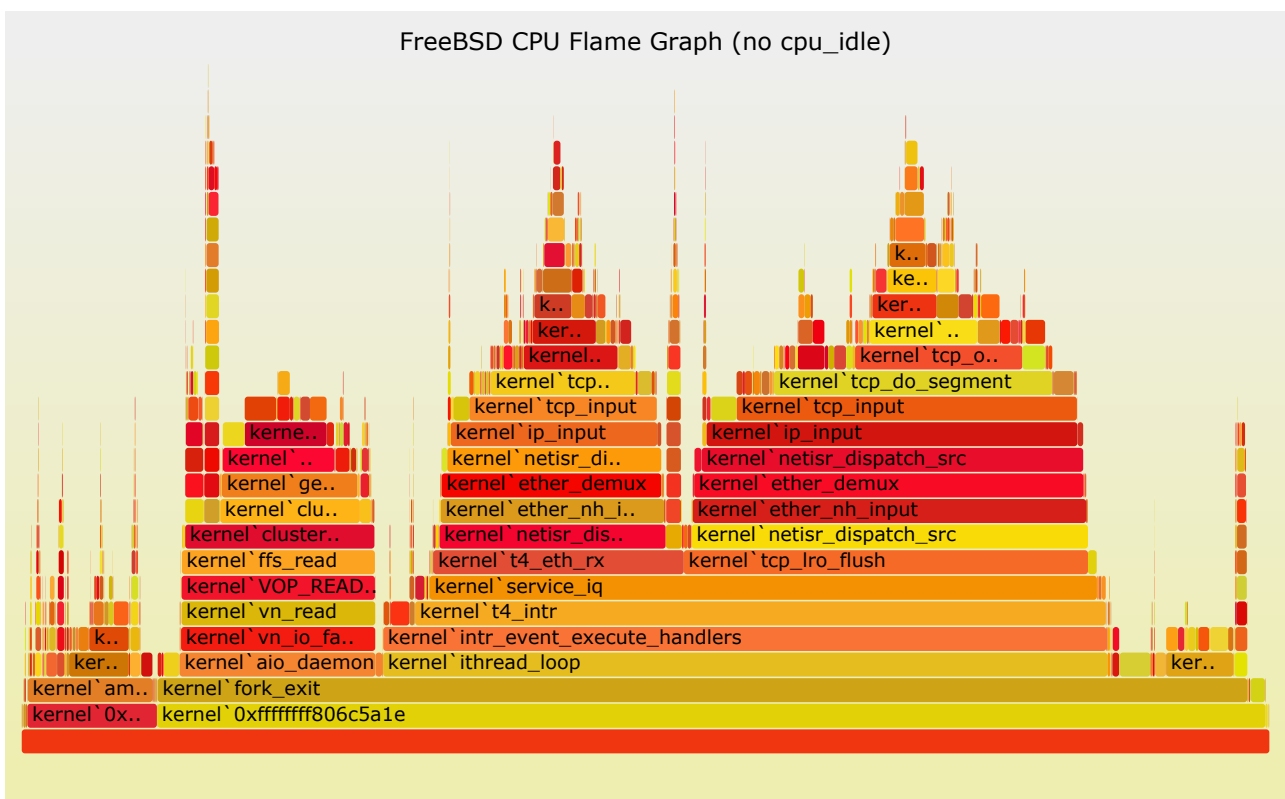
(This is the first time I've published a screenflow. What do you think? This might be better than nothing for my demos that aren't otherwise recorded. Thanks to Deirdré Straughan for help editing this.)

We're using flame graphs on the Netflix Open Connect Appliances (OCAs) to understand CPU usage and look for optimizations. Not just CPU flame graphs, but also [CPI flame graphs](#), where the color indicates cycles-per-instruction (CPI). This shows what the CPUs are *really* doing: are they busy retiring instructions (executing code), or are they stalled on memory I/O?

Flame graphs are easy to generate given the output of a profiler (DTrace or pmcstat), and I included the commands in the talk. For example, to generate a kernel CPU flame graph using DTrace, by sampling kernel stacks at 197 Hertz for 60 seconds:

```
# git clone https://github.com/brendangregg/FlameGraph      # or download zip
# cd FlameGraph
# kldload dtraceall      # if needed
# dtrace -x stackframes=100 -n 'profile-197 /arg0/ {
#     @[stack()] = count(); } tick-60s { exit(0); }' -o out.stacks
# ./stackcollapse.pl out.stacks | ./flamegraph.pl > out.svg
```

Here's an example resulting flame graph ([SVG](#)). Click to zoom:



And to generate a stall cycle flame graph using pmcstat, by sampling stacks based on resource stalls:

```
...  
# pmcstat -S RESOURCE_STALLS.ANY -O out.pmcstat sleep 10  
# pmcstat -R out.pmcstat -z100 -G out.stacks  
# ./stackcollapse-pmc.pl out.stacks | ./flamegraph.pl > out.svg
```

Many other types are possible, as I covered in the talk.

We've automated CPU flame graph generation on the OCAs (thanks, Scott), so that they can not only be created easily, but also used for non-regression analysis.

I had done an epic flame graphs talk at USENIX/LISA 2013 ([slides](#), [video](#)) lasting 90 minutes. It was really two talks in one: CPU flame graphs, then other flame graph types. I was thorough and explained everything, but I've since hesitated to encourage people with casual interest to watch a 90 minute video. For the FreeBSD dev summit, I gave a tour of all flame graph types in 50 minutes, and really got stuck into the technical details fast, skipping introductions. It was a lot of fun, and perhaps I should be doing more talks like this.

This [FreeBSD Developer and Vendor Summit](#) talk was a day after my [performance analysis](#) talk at MeetBSD CA, which I also recommend watching.

In a follow up post, I'll discuss the new method for creating off-CPU flame graphs using procstat that I demoed during the talk.

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).