

A New, Static, Blog

20 Apr 2014

I've been having fun playing with [Jekyll](#), a static blog engine. These build HTML files for serving directly, rather than dynamically generating them on-demand (as is often the case) using PHP and MySQL. This blog is powered by Jekyll. (Yes, I know about [Octopress](#), which can automate this further.)

Static blog engines have high performance: there is no latency for page requests, which for some interpreters can take tens of milliseconds. There is also no runtime CPU, memory, or storage overheads for the blog engine software and database. Also, since the pages are static, they will cache easily.

One night I debugged a friend's broken Wordpress site, where pages mysteriously stopped loading part way through. Out of habit I checked the system log, and found entries that beggared belief. The cloud provider was killing PHP, and noting that it had in the system log. Something like:

```
hey, I killed your process named "php", PID 8164, CPU exceeded
hey, I killed your process named "php", PID 8165, CPU exceeded
[... etc ...]
```

It turned out that the PHP page generation time exceeded an imposed CPU limit by the cloud provider, who sent a KILL signal to the PHP process *before it could emit a single page*. Think [OOM killer](#) for CPU. While I find that approach A LITTLE CRAZY – given that this is a time sharing system and the different ways to tune the scheduler to achieve a similar but more GENTLE result – I suppose there is some merit here: the ability to kill run-away processes. That is, if the threshold is set to a long interval (minutes) and not what it was in this case (milliseconds)!

This is my personal blog, where I'll talk about random things, mostly related to computer performance, with some tech obscurity and hobbies. I don't expect anyone to really read this, rather, people will find useful bits and pieces later as a result of an Internet search. People including me.

You may know me from prior blogs: [bdgregg.blogspot.com](#) (2005-6), [blogs.sun.com/brendan](#) (2006-2010) which became [blogs.oracle.com/brendan](#) after the acquisition, and [dtrace.org/blogs/brendan](#) (2010-2014). If I work on DTrace some more (I'm not [expecting to](#)), I'll try to share it on the [dtrace.org](#) blog as well.

There are plenty of exciting things I'm doing as a senior performance engineer at Netflix, and I hope to share them on the [Netflix tech blog](#), especially when it's a collaboration with other Netflix staff.

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).

This Site:

[Homepage](#)

[Blog](#)

[Full Site Map](#)

[Sys Perf book](#)

[Linux Perf](#)

[Perf Methods](#)

[USE Method](#)

[TSA Method](#)

[Off-CPU Analysis](#)

[Active Bench.](#)

[Flame Graphs](#)

[Heat Maps](#)

[Frequency Trails](#)

[Colony Graphs](#)

[perf Examples](#)

[eBPF Tools](#)

[DTrace Tools](#)

[DTraceToolkit](#)

[DtkshDemos](#)

[Guessing Game](#)

[Specials](#)

[Books](#)

[Other Sites](#)