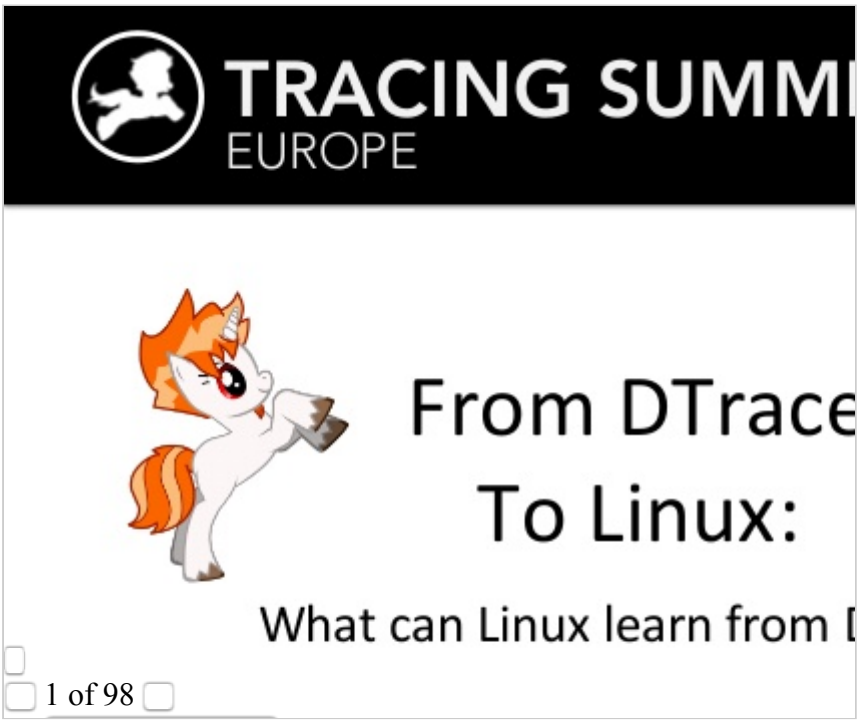
# Tracing Summit 2014: From DTrace To Linux

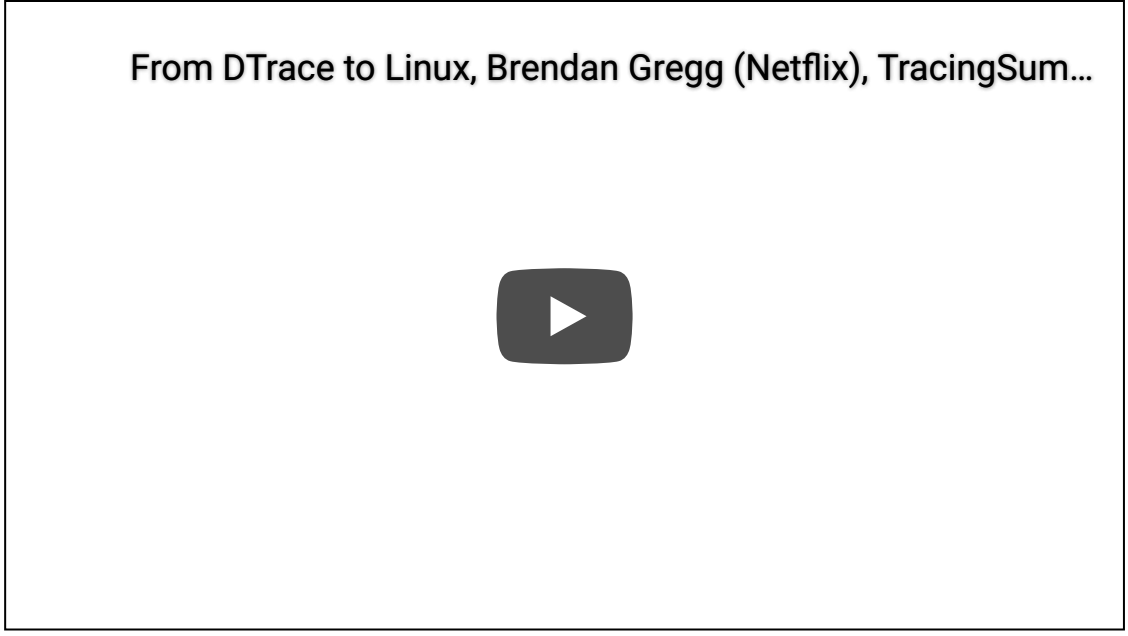
28 Feb 2015

What can Linux learn from DTrace: what went well, and what didn't go well, on its path to success? This was the topic for my talk at the Tracing Summit late last year in Düsseldorf, where I discussed not just the DTrace technology, but lessons from marketing and adoption of a system tracer, and an inside look at how DTrace was really deployed and used in production environments.

The slides are on slideshare:

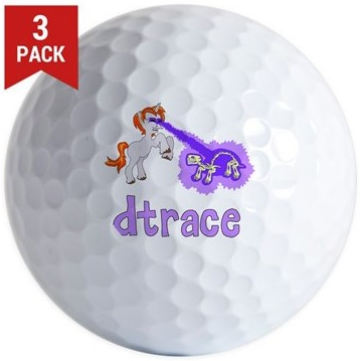



The talk was also videoed, which is on youtube:




I addressed the key question: why doesn't Linux have DTrace?

One way answer to this is to look at ftrace and perf_events, which are Linux technologies built-in to the kernel, and provide many similar capabilities to DTrace. These technologies didn't exist when DTrace was launched in 2005, when everyone learned not only that DTrace existed, but that it created an observability gap for Linux. But it's no longer 2005: the observability gap is now much smaller and shrinking. Linux is catching up.




But there's another answer to this, which is an important and previously untold story that Linux needs to learn. DTrace is more than just a great technology: DTrace had marketing. DTrace was pushed by sales folk. DTrace had training classes, and instructors. DTrace had evangelists. DTrace had user-friendly scripts (many of which I wrote). DTrace had a community.

By comparison, technologies like ftrace and perf_events in Linux are so far little more than just the technologies, documented in the kernel tree. No Linux backer is spending big to market them. And why should they? Sun Microsystems, the creator of DTrace, was fighting for its survival. It could justify spending millions marketing and selling its differentiators, including DTrace. I called it the "trace race". I knew many of the Sun staff involved – and they did a heck of a job (thank you). Such a scenario may never be repeated.

Marketing serves a role: it helps people become aware of technologies, and motivates them to try them out. I know many Linux users who didn't know about ftrace until I started blogging and speaking about it. And that's what we as a community need to do: use the technologies, and share the use cases. We won't have the backing of corporate marketing and sales, but we can use social media.

There's many more lessons in the talk, which I recommend watching. It's my only talk where I covered more than just technology, but many non-technical factors involved in a product's success.

I've been contributing to Linux tracing myself, by participating in the communities (eg, linux-perf-users mailing list), testing new features (eg, eBPF), publishing tools and documentation (eg, perf-tools, and blog posts here), and talking about these technologies at conferences. This includes my recent Linux Profiling at Netflix talk at SCALE13x, where I covered perf_events, and my LISA 2014 talk, where I covered my mostly ftrace-based perf-tools. I'll be posting more on this blog about Linux tracing during the year.

Slides and video for all the Tracing Summit talks are on the Tracing Summit wiki (thanks to the EfficiOS folk for filming them). A couple of my favorites were Yannick Brosseau's "Using tracing at Facebook scale", and Francis Giraldeau's "Discover what your app is waiting with kernel tracing". Thanks to the organizers for putting on a great event! (And if you watched the talk, yes, Deirdré Straughan helped me a little with this post as well.)

*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).*


About this blog

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