Working at Netflix 2017

16 May 2017

I've now worked at Netflix for over three years. Time flies! I previously wrote about Netflix in 2015 and 2016, and if you are interested in what it's like to work here, I already covered much in those posts. As before, no one at Netflix has asked me to write this, and this is my personal blog and not a company post.

I'll start with some exciting news, describe what my job is really like, the culture and mission, and some work updates.

100 Million Subscribers!

When I joined Netflix in April 2014, we had over 40 million subscribers in 41 countries. We are now in 190 countries and just crossed 100 million subscribers! It's been thrilling to be part of this and help Netflix scale.

You might imagine that at some point we had a major scaling crises, where it looked like we'd fail due to an architectural bottleneck, and engineers worked long nights and weekends to save Netflix from certain disaster. That'd make a great story, but it didn't happen. We're on the EC2 cloud, which has great scalability, and our own cloud architecture of microservices is also designed for scalability. During this time we did do plenty of hard work, rolling out new technologies and major microservice versions, and fixed many problems big and small. But there was no single crisis point. Instead, it has been a process of continual improvements, by many engineers across the company.

A Day in the Life (Performance Engineering)

What do I actually do all day?

Most of my day is a 50/50 mixture of proactive projects, and reactive performance analysis. The proactive projects usually take weeks or months, and are where I'm developing a new technology or helping other teams with performance analysis or evaluations. Most of these projects aren't public yet, and some of them involve working with other companies on unreleased products. My work with Linux is different in that it is mostly public, and includes my perf-tools and bcc/eBPF tracing tools.

Another long term project is Vector, our instance analysis tool, where I'm adding new performance analysis features. Getting frame pointer support in Java was another project I did a while ago.

The reactive work can be for any performance problem that shows up, involving runtimes (Java, Node.js), Linux (and sometimes FreeBSD), or hypervisors (Xen, containers). Recently that's included:

- Debugging why perf profiling stopped working in recent Docker containers.
- Java core dump analysis for a crashing JVM.
- MSR analysis on a instance to show it was running at a lower clock rate.
- A latency outlier issue that happened every 15 minutes.

- Analyzing slab memory growth on a instance with containers.
- Getting flame graphs to work in a new environment.

Staff ask for help over chat, either to the perfeng chatroom or me directly, or they come visit my desk in F2. I'm also monitoring various chatrooms and metrics, and will jump in when needed.

It's a good balance. Too much reactive work and you don't have time to build better tools and general fire proofing. Too much proactive work and you can become disconnected from the current company pain points, and start building solutions to the problems of yesteryear.

About one hour on average each day is meetings. Some of these are regular meetings: we have a team meeting once every two weeks where everyone discusses what they are working on, and I have a one-on-one with my manager once every two weeks. At a lower frequency, I have scheduled meetings with my manager's manager, and their manager. All these manager meetings keep me informed of the current company needs, and help connect me to the right people and projects at Netflix.

Once every two weeks, I summarize what I've been working on in a shared doc: the team's bi-weekly status.

Then there's some random events that happen during the year. We have offsites, where we plan what to work on each quarter, and team building events. There's also unofficial recreational groups at Netflix, including movie clubs (for good movies, and for bad ones), a karaoke group (which includes some Hamilton fans), and various sports teams. I'm on the Netflix cricket team (if you're at Netflix and didn't know we exist, join the cricket chatroom). I also usually speak at some conferences each year.

Culture

The biggest difference I've found working here is still the culture. We are empowered to do the right thing, and believe in "freedom and responsibility". This is documented in the Netflix culture deck, and after three years I still find it true.

The first seven slides point out that companies can have aspirational values, but the actual values differ:

> The *actual* company values, as opposed to the *nice-sounding* values, are shown by who gets rewarded, promoted, or let go

Before joining Netflix, you're told to read it and see if this company is right for you. Then while working here, staff cite the culture deck in meetings for decision making advice. It's not nice-sounding values that are printed in the lobby and people forget about. It's an ongoing influence in the day to day running of Netflix. Having it online also beats learning the culture through word of mouth or trial and error.

I know people in tech who are burned out but stay in lousy jobs, assuming every workplace is just as terrible. Jobs where there is little to no freedom, no responsibility or accountability, and where dumb office politics is the norm. I wish everyone could have a chance to work at a company like Netflix. Little to no bureaucracy. You can focus on engineering and getting stuff done, with awesome staff who will help you.

Mission

I spoke about this in my 2015 post, but it's worth repeating: our mission is to improve how entertainment is consumed worldwide, by building a great product that people choose to buy.

I've noticed a widespread cynicism about successful companies, especially US corporates, where it's assumed that they must be doing something shady to be really competitive. Like selling customer data, or making it difficult to terminate membership. It's been amazing and inspiring to see how Netflix operates, contrary to this belief. We don't do anything shady, and we're proud of that. We're an honest company.

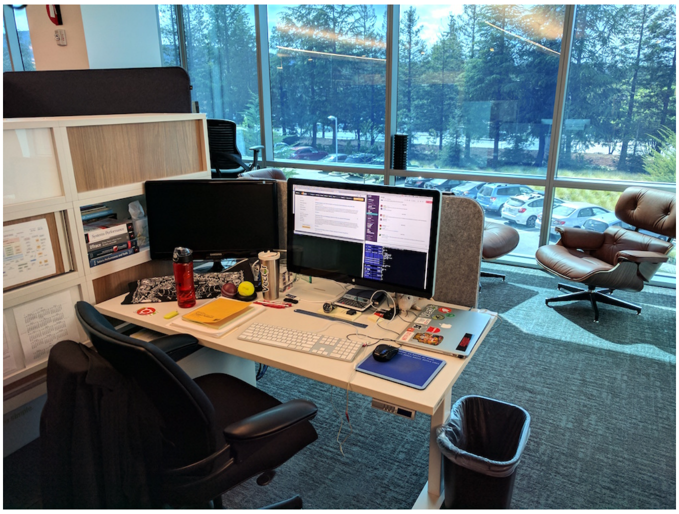
Work Updates

SRE: Last year I talked about my site reliability engineering (SRE) work. Since then, our CORE SRE team has grown and I'm no longer needed on the on-call rotation, so I'm back to focusing on performance work. My 18 months of SRE on-call provided many memories and valuable experiences, as well as a deeper understanding of SRE. I talked about what I learned in my SREcon 2016 keynote, and how the aims and tools differ between performance engineering and SRE performance analysis.

I miss the thrill of being paged and knowing I'm going to work with other awesome engineers and fix something *important* in the next five minutes... or at least try to! If I miss this thrill too much, I can always jump into the CORE chatroom and help with production issues when they happen.

Linux: I've been contributing to profilers and tracers, and it's been satisfying to help fix these areas that I really care about. In the last three years I developed the ftrace-based perf-tools and used them to solve many problems, which I wrote about in lwn.net and spoke about at LISA 2014. I also worked with Alexei Starovoitov (now at Facebook) on enhanced BPF for tracing, and developed many bcc tools that use BPF. I spoke about these at Facebook's Performance@Scale event and other conferences. We're rolling out newer kernels now, and it's pretty exciting to use my bcc tools in production.

For Linux, I've also done tuning, kernel analysis, gdb, testing of hist triggers, testing of some perf patches, and contributed a few trivial patches of my own.



My new desk in building F

PMCs: When I considered joining Netflix three years ago, I had two technical concerns: 1. No advanced Linux tracer, and 2. No PMC access in EC2. How am I going to do advanced analysis without these? The more I thought about it, the more I became interested in the challenge, which would be the biggest of my career. Three years later, I've helped solve both of these (as well as devise some workarounds along the way). Now we have Linux 4.9 eBPF and The PMCs of EC2. Thanks to everyone who helped.

Team Changes: Our team has grown a little, and we have a new manager, Ed Hunter, who I worked for before at Sun Microsystems. It's great to be working with Ed again. Our prior manager, Coburn, was promoted.

Summary

When I use an awesome technology, I feel compelled to post about it and share. In this case, it's an awesome company and culture. After three years, I still find Netflix an awesome place to work, and every day I look forward to what I'll work on next.

Update: Many people have been emailing me their resumes. I'm glad they are interested in working at Netflix, but I'm an engineer, not a hiring manager. Please use jobs.netflix.com, which will send your details to the right people.

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).