Netflix and FreeBSD: Using Open Source to Deliver Streaming Video
Open Connect
Open Connect is Netflix’s CDN. It is global, efficient, and purpose-built for distributing Netflix’s content.
The Open Connect Appliance

The OCA is the “backbone” of the Open Connect network.

The OCA almost exclusively runs open-source software.

40Gb/s Storage Appliance with 248TB storage (2RU form factor)
Open Connect Traffic

Video Apps

Video Apps

Video Apps

Open Connect

Video Apps

Video Apps

Video Apps

Video Apps
Netflix OCA Workload
Using FreeBSD and commodity parts, we achieve 90 Gb/s serving TLS-encrypted connections with ~55% CPU on a 16-core 2.6-GHz CPU.
OCA Operating System
(Abridged) BSD “Family Tree”

AT&T Unix

BSD

NetBSD

FreeBSD
We Track FreeBSD “Head”

FreeBSD head

Netflix master

Netflix release branches
Typical Release Cycle

- Five Weeks of Development
- Five Weeks of Testing/Deployment

- FreeBSD Merge
- Feature Development/Integration
- Testing
- Dev Testing
- Canary Testing
- Phased Rollout
Examples of Features

- NUMA enhancements
- Asynchronous sendfile
- Kernel TLS
- Pbuf allocation enhancements
- “Unmapped” mbufs
- I/O scheduling
- TCP algorithms
- TCP logging infrastructure
Tracking “head” lets us stay forward looking and focused on innovation.
Downstream users of open-source projects can be stuck in “vicious” or “virtuous” cycles.
Vicious Cycle

Infrequent Merges

Many Conflicts/Regressions

Slower Feature Velocity
Virtuous Cycle

Frequent Merges

Faster Feature Velocity / Collaboration

Few Conflicts / Regressions
Reasons We Keep Local Diffs

- Information covered under NDA
- Feature which is still in development/testing
- Feature which needs to be generalized
It is our intention to upstream any code which we can.
Benefits to Netflix of Tracking FreeBSD “Head”

- Quicker feature iteration
- Quicker access to new FreeBSD features
- Quicker bug fixes
- Enables collaboration
- Minimizes merge conflicts
- Amortizes merge “cost”
Benefits to FreeBSD

- Wide deployment of “head” branch code (albeit in a narrow use case)
- Early intensive testing
- Incentive for Netflix to upstream code
Objections to Running “Development” Code

- It isn’t stable
- Why should you pay to find the bugs others will find while testing head?
- Aren’t there more security bugs?
- No one runs development branches
- Pay monthly “cost” to do merges
- You get new bugs each month
Running FreeBSD “head” lets us deliver large amounts of data to our users very efficiently, while maintaining a high velocity of feature development.
Thank you