Netflix and FreeBSD: Observations from ~3 Years of Running Head
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 workhorse 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

Video Apps
Netflix OCA Workload

RAM

Plain-text Data

Encrypted Data

Disks

CPU

NICs
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
Why do we use FreeBSD?

We came for the license. We stay for the efficiency.
FreeBSD Release Cycle

- head
- stable/11
- 11.0
- 11.1
- 11.2
- stable/12
- 12.0
- 12.1

NETFLIX AND FREEBSD
BSDCan 2019
We Track FreeBSD “Head”

FreeBSD head

Netflix master

Netflix release branches
**Typical Release Cycle**

Five Weeks of Development

- FreeBSD Merge
- Feature Development/Integration

Testing

Five Weeks of Testing/Deployment

- 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.
Observations from Running Head

- Easier to collaborate with others
- Faster bug fixes and features
- Easier to upstream code
  - Also better, as what we upstream is the same code we run internally
- When tracking head, upstream code freezes are more disruptive than helpful
- API/KPI changes are easy to handle
- ABI/KBI changes are (mostly) a non-issue
- Head quality is so high that bug fallout is manageable
Benefits to the FreeBSD Projects

- 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