# Netflix CDN and Open Source

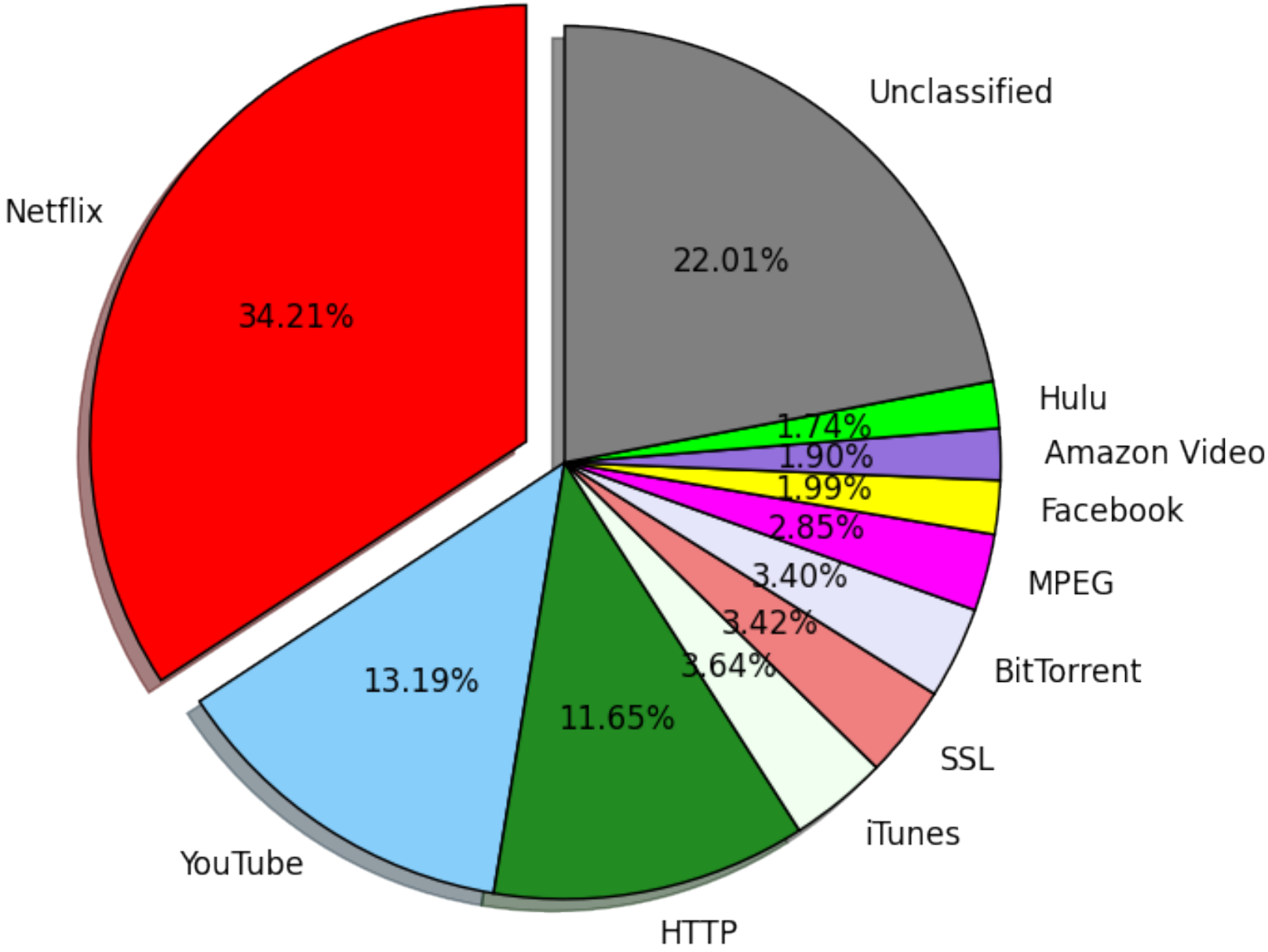
Gleb Smirnoff
glebius@nginx.com

Nginx.conf 2014
San Francisco

October 21, 2014

# Netflix in numbers

- 50+ millions subscribers in 40+ countries
- Over 1 Petabyte of data




2014 1H Sandvine North Amertica Traffic Report

# Components of Netflix Streaming

- Amazon Web Services
  - Website, Business Logic, Customer Authentication
  - Data Sciences, Encoding/Encryption
  - Command and Control
- Content Streaming
  - ~~Big 3 CDN's (Akamai, Limelight, Level3)~~
  - Own CDN

# Reasons for running own CDN

- Grow faster
- Reduce costs

[1] https://www.netflix.com/openconnect

# Reasons for running own CDN

- Grow faster
- Reduce costs
- Control the server side of HTTP connection
- Build a CDN specialized in Netflix content delivery
- Put the content closer to a client

[1] https://www.netflix.com/openconnect

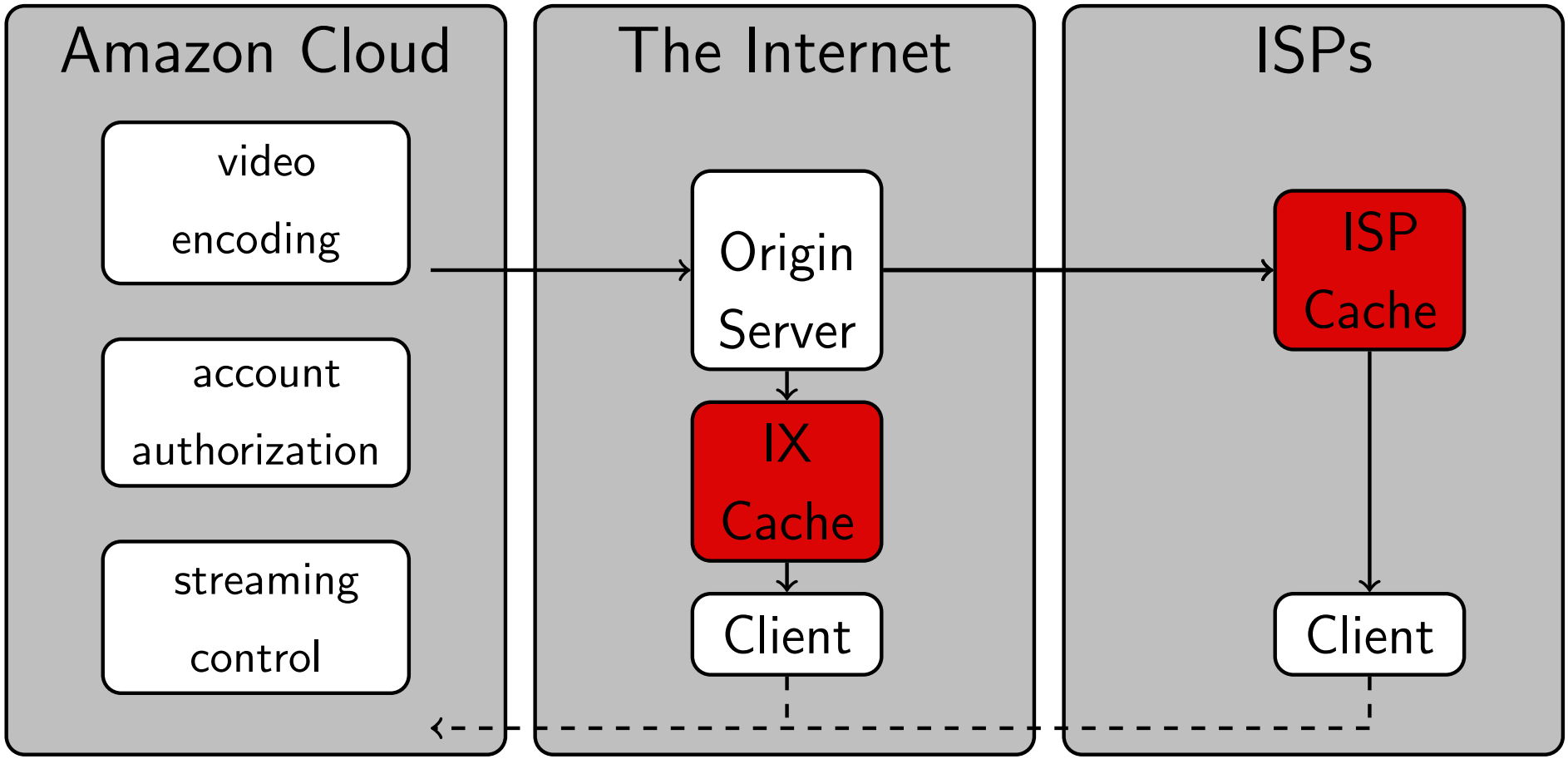
# Reasons for running own CDN

- Grow faster
- Reduce costs
- Control the server side of HTTP connection
- Build a CDN specialized in Netflix content delivery
- Put the content closer to a client

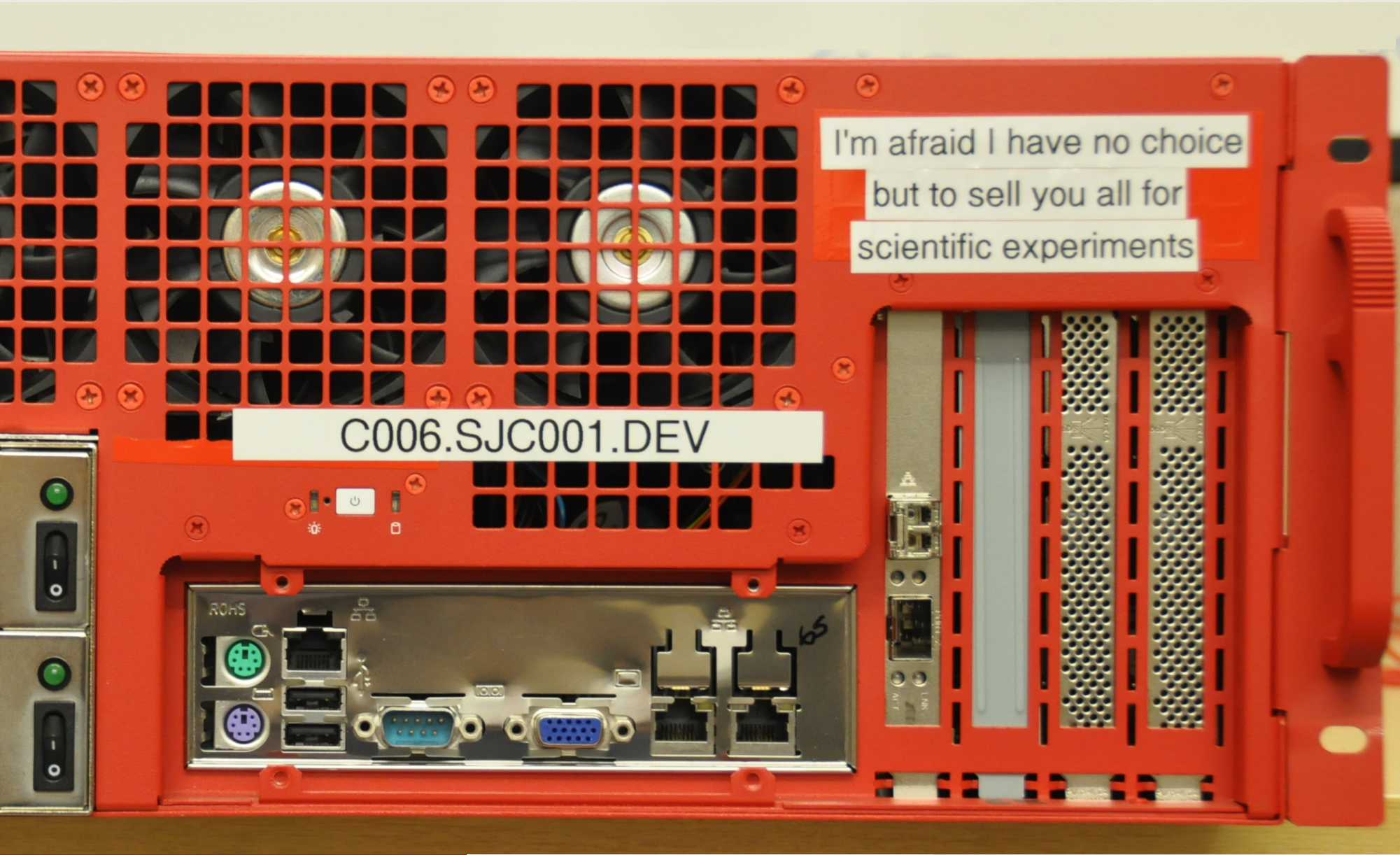
Solution: Open Connect[1]

[1] https://www.netflix.com/openconnect

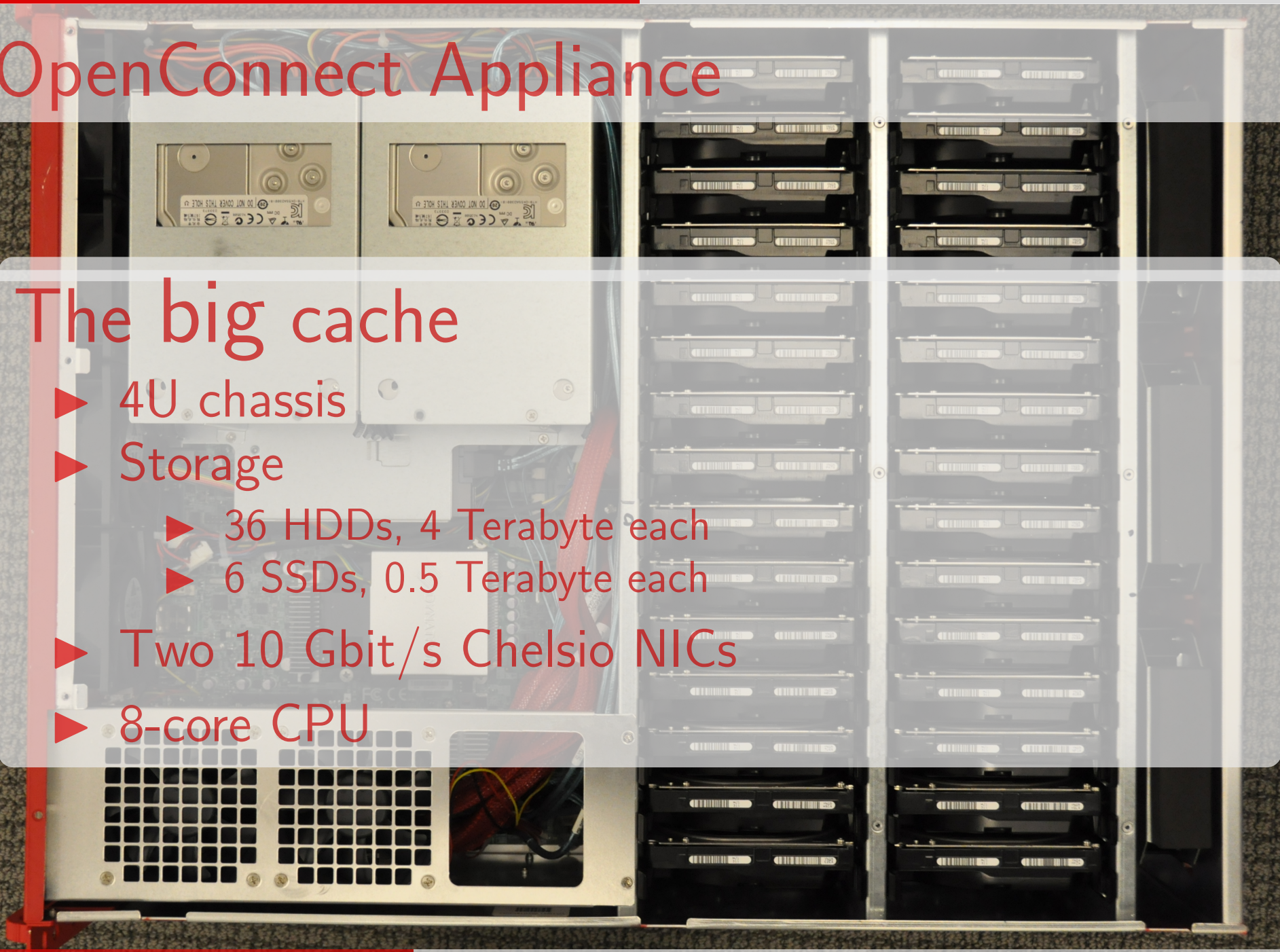
# The OpenConnect Initiative



# OpenConnect Appliance



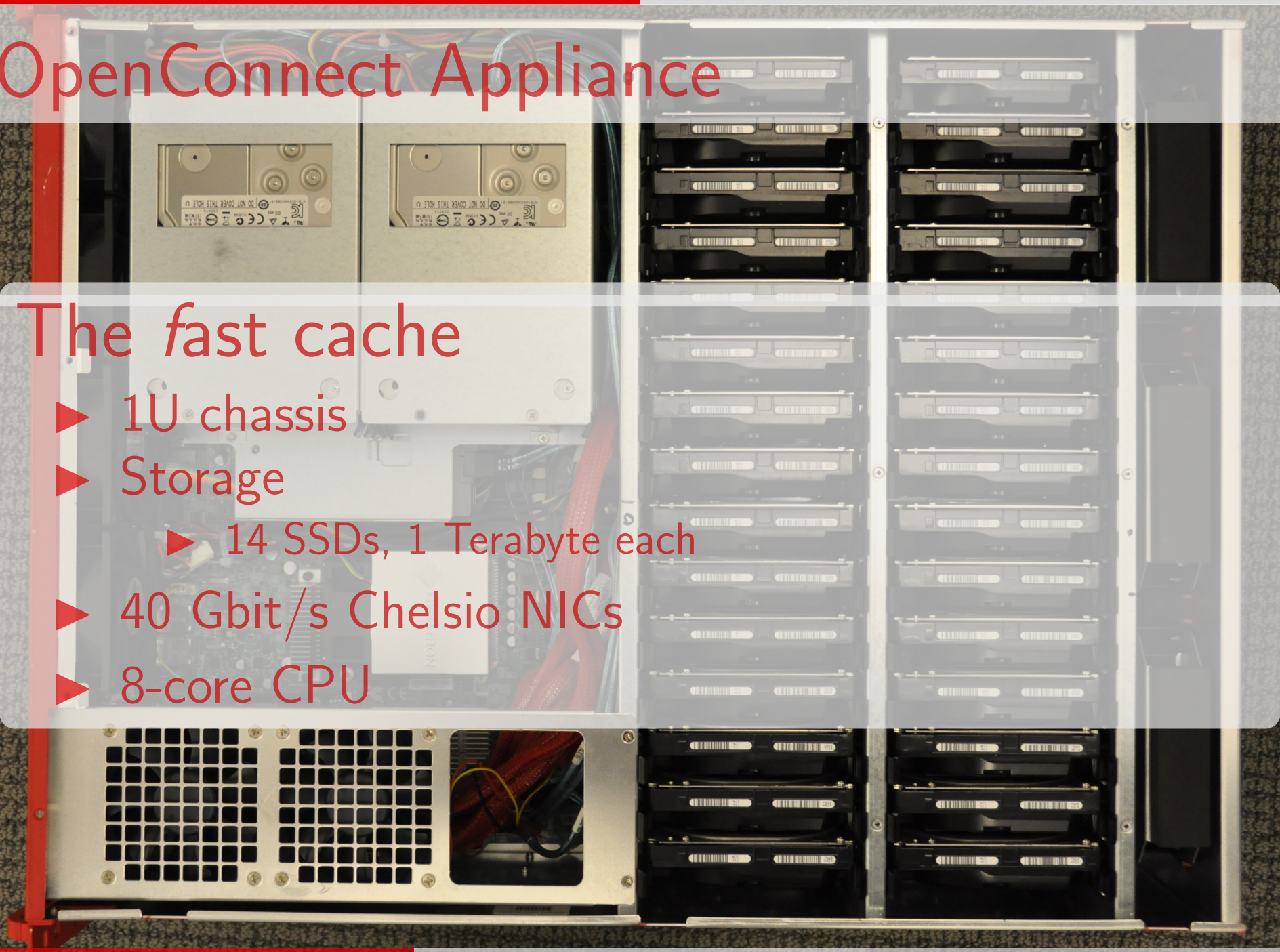
# OpenConnect Appliance



## The big cache

- 4U chassis
- Storage
  - 36 HDDs, 4 Terabyte each
  - 6 SSDs, 0.5 Terabyte each
- Two 10 Gbit/s Chelsio NICs
- 8-core CPU

# OpenConnect Appliance



## The *fast* cache

- 1U chassis
- Storage
  - 14 SSDs, 1 Terabyte each
- 40 Gbit/s Chelsio NICs
- 8-core CPU

# OpenConnect Appliance software

- FreeBSD operating system
  - nanobsd(8) based firmware
  - some enhancements
- NGINX web server
  - custom modules
- BIRD routing daemon

# Software choice

- Open source
- BSD license

# Software choice

- Open source
- BSD license
- FreeBSD
  - Known to be fast and stable
  - Strong developer community

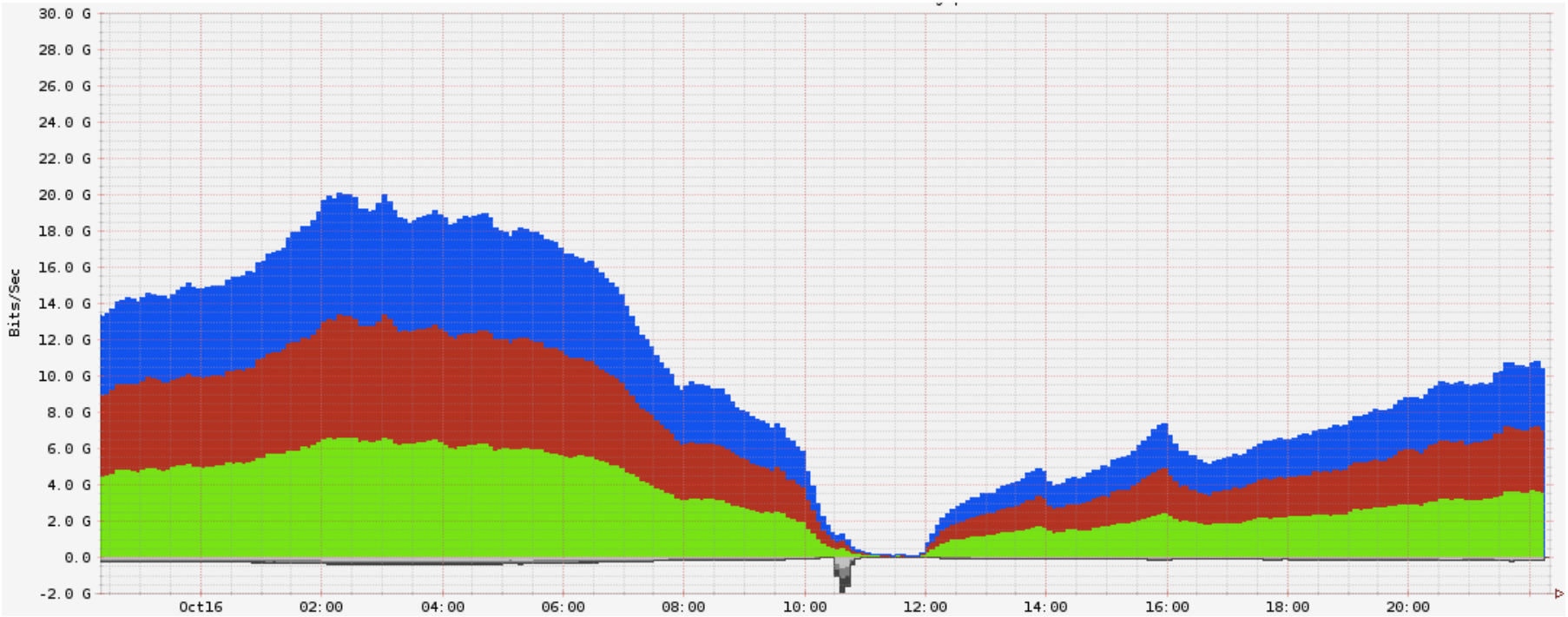
# Software choice

- Open source
- BSD license
- FreeBSD
  - Known to be fast and stable
  - Strong developer community
- NGINX
  - Known to be fast and stable
  - Commercial support from Nginx, Inc.
  - Flexible framework for custom modules

# Software choice

- Open source
- BSD license
- FreeBSD
  - Known to be fast and stable
  - Strong developer community
- NGINX
  - Known to be fast and stable
  - Commercial support from Nginx, Inc.
  - Flexible framework for custom modules
- FreeBSD + NGINX
  - kqueue(2) event driven model
  - sendfile(2) + aio(4) mode

# OpenConnect operation: serve & fill



# OpenConnect: streaming numbers

- 400,000 stream files per appliance
- 5000-30,000 client streams per appliance
- 300-1000 clients per disk
- Clients request 80Kb-300Kb ranges

# Dealing with open source

The traditional way:

- Take a stable, well tested version, fork it
- Develop your product on top of it

# Dealing with open source

The traditional way:

- ~~Take a stable, well tested version, fork it~~
- ~~Develop your product on top of it~~

# Dealing with open source

The Netflix rules:

- Pull the bleeding-edge version of software
- Push your changes upstream aggressively

# Myths about development version

Myth #1:
Development version is full of bugs

# Myths about development version

Myth #1:

~~Development version is full of bugs~~

---

Truth: most bugs are discovered in stable versions

# Myths about stable version

Myth #2:
We will wait for a stable version, and someone else will fix bugs [2]

[2] http://en.wikipedia.org/wiki/Free_rider_problem

# Myths about stable version

Myth #2:

~~We will wait for a stable version, and someone else will fix bugs~~ [2]

---

Truth: No one will discover *your* bugs

[2] http://en.wikipedia.org/wiki/Free_rider_problem

# Discovering bugs in different versions

| development | stable |
| --- | --- |
| code is “hot” | code is unmaintained |
| no API/ABI constraints | API/ABI must be preserved |

# More myths

Myth #3:
Not following development version saves us a lot of time

# More myths

Myth #3:

~~Not following development version saves us a lot of time~~

---

Truth: some day you will need to go forward

# Myths on open source

## Myth #4:
## Sharing code discloses know-how's

# Myths on open source

## Myth #4:
## ~~Sharing code discloses know-how's~~

---

Truth: know-how's reside in a tiny percent of code, or even outside

# Noble or selfish?

- We want to influence the direction of open source development
- We want to outsource maintainance of our code to community
- We want more eyes to examine our code
- We want more people to discover bugs in it

# Noble or selfish?

- We want to influence the direction of open source development
- We want to outsource maintainance of our code to community
- We want more eyes to examine our code
- We want more people to discover bugs in it
- And, of course, we want to be considered noble givers of code to community ☺

# OpenConnect performance

- OpenConnect started in 2011
- $<$ 10 Gbps per appliance

# OpenConnect performance

- OpenConnect started in 2011
  $<$ 10 Gbps per appliance
- Now, in 2014, we achieve
  $>$ 30 Gbps per appliance

# OpenConnect performance

- OpenConnect started in 2011
  $<$ 10 Gbps per appliance
- Now, in 2014, we achieve
  $>$ 30 Gbps per appliance
- Next goal is $>$ 80 Gbps per appliance

# Areas of focus

- network stack: sockets, TCP, drivers
- storage: drivers, UFS
- VM subsystem: caching

# Performance improvements

- Reducing lock/cache line contention in
  - link aggregation driver
  - kernel socket buffers
  - sendfile(2) kernel memory buffers

# Performance improvements

- Reducing lock/cache line contention in
  - link aggregation driver
  - kernel socket buffers
  - sendfile(2) kernel memory buffers
- Reducing complexity, structuring code in
  - kernel flowtable
  - NGINX sendfile(2) code

# Performance improvements

- Reducing lock/cache line contention in
  - link aggregation driver
  - kernel socket buffers
  - sendfile(2) kernel memory buffers
- Reducing complexity, structuring code in
  - kernel flowtable
  - NGINX sendfile(2) code
- Introducing multithreading in
  - UFS softupdates

# Various improvements & bugfixes

- NGINX core and modules
- IPv6 network stack
- UFS journaling

# Completely new features

- Per-CPU statistical counters:
  - Precise: do not lose updates
  - Fast: do not use any locking, neither critical sections

# Completely new features

- Per-CPU statistical counters:
  - Precise: do not lose updates
  - Fast: do not use any locking, neither critical sections
- Completely new sendfile(2) implementation:
  - Doesn't block on disk I/O!
  - Allows to specify readahead
  - Allows to deny VM caching

# Working with community

- unmapped I/O
- VM radix

# Future and work in progress

- cc_netflix: new TCP congestion control algorithm
- hardware assisted TCP pacing
- kernel-side TLS offload
- SSD I/O performance improvements
- multithreading pagedaemon
- NUMA support

# Questions?