# NETFLIX | OPEN CONNECT

BY GREG WALLACE

## Overview

Netflix (NASDAQ: NFLX) is the world's leading streaming entertainment service with 183 million paid memberships in over 190 countries enjoying TV series, documentaries and feature films across a wide variety of genres and languages. Members can watch as much as they want, anytime, anywhere, on any internet-connected screen. Members can play, pause and resume watching, all without commercials or commitments. www.netflix.com

Open Connect is the name of the global network that is responsible for delivering Netflix TV shows and movies to members world-wide. This type of network is typically referred to as a Content Delivery Network, or CDN, because its job is to deliver internet-based content (via HTTP/HTTPS) efficiently by bringing the content that people watch close to where they're watching it. Open Connect Appliances run a lightly customized version of FreeBSD. https://openconnect.netflix.com/Open-Connect-Overview.pdf

Netflix employs several FreeBSD committers and additional members of the Open Connect team also contribute code upstream.

**INDUSTRY**

**STREAMING ENTERTAINMENT SERVICE**

**LOCATION**

**HEADQUARTERS IN LOS GATOS, CALIFORNIA**

**EMPLOYEES**

**6700 WORLDWIDE**

## Open Connect Pushes Over 100 Tb/s Peak

Those of us old enough to remember the dot com and telecom boom may recall the emblematic 1999 Quest Communications advertisement in which a weary traveler checks into a hotel in the middle of nowhere. The clerk promises a lackluster breakfast, but entertainment?

**That** they have in spades. "Every movie ever made, in any language, anytime day or night."

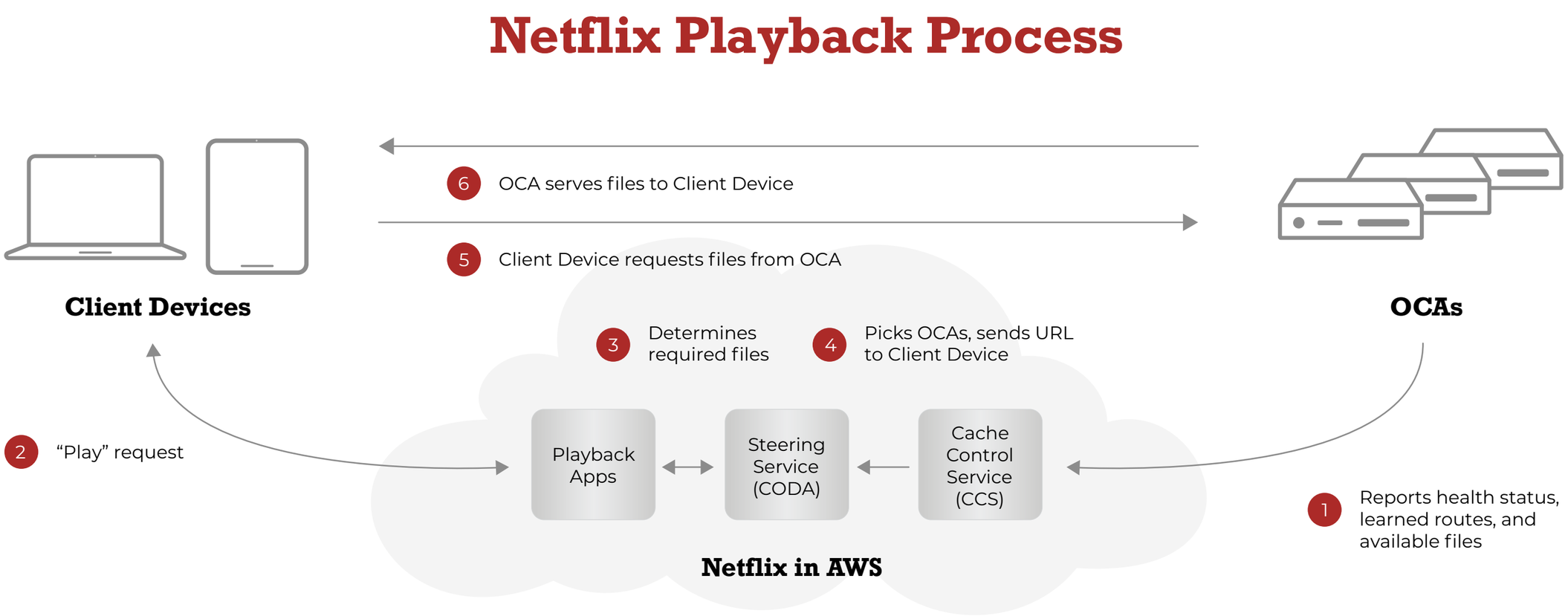
Flabbergasted, the guest wonders aloud "how is that possible?" How indeed! (read on). Twenty years later, and hotel TVs are some of the last devices to provide every movie ever made. Technology, it seems, is not without a sense of irony.

No discussion of the latest trends in streaming entertainment and the technology that makes it possible is complete without Netflix. As of April 2019, the Netflix U.S. catalog consisted of 47,000 TV shows and 4,000 movies. Netflix reports that the global Open Connect Network pushes over 100 Tb/s of traffic at peak. According to Sandvine, this represented about 15% of total internet traffic in 2019.

## Open Connect: A Network And A Program

Netflix began the Open Connect initiative in 2011 as a response to the ever-increasing scale of Netflix streaming. Two primary reasons motivated the program:

1. As Netflix grew to be a significant portion of overall traffic on consumer Internet Service Provider (ISP) networks, it became important to be able to work with those ISPs in a direct and collaborative way
2. Creating a content delivery solution customized for Netflix allowed their engineers to design a proactive, directed caching solution that is much more efficient than standard demand-driven CDNs. The directed caching architecture reduces the overall demand on upstream network capacity by several orders of magnitude.




## The Network

Most CDNs work in what's called a demand-driven way. This means that *what* the network caches and *where* is determined by what is requested in a particular area. For general purpose CDNs where there is limited ability to predict the content people will want, this works well.

Because Netflix controls the end user apps and has detailed information about viewing trends, they could achieve significant efficiencies moving to a directed CDN. In the Netflix directed CDN model, their fleet of Open Connect Appliances (OCAs), described in detail below, receive daily catalog updates during what are called Fill windows when viewing is very low.

## The Program

Netflix has an open peering policy, meaning they will peer with any ISP that agrees with the terms of the program. Open peering improves internet user experience by localizing traffic. It also has the advantage of reducing transit costs, a benefit to Netflix, ISPs, and the internet as a whole.

In addition to OCAs in Netflix data centers and installed in Internet Exchange Points (IXPs), Netflix provides OCAs free of charge to qualifying ISPs for installation directly in the ISP's network. This increases localization and reduces upstream traffic even further.[1] Interestingly, the fact that these OCAs are owned by Netflix, but used by the ISP, raised some licensing considerations that initially drew the Open Connect engineers to FreeBSD for its permissive license.[2]

1 See https://openconnect.netflix.com/Open-Connect-Overview.pdf for program information.
2 https://www.nginx.com/blog/why-netflix-chose-nginx-as-the-heart-of-its-cdn/

## Open Connect Appliances

The workhorses of the Open Connect CDN are the Open Connect Appliances, or OCAs for short. These appliances, of which there are three primary configurations, run a lightly customized version of FreeBSD head, or development, branch. That such a large and mission critical network would run the fast-moving development branch may at first blush seem risky. At the 2019 FOSDEM conference, Jonathan Looney, Netflix Engineering Manager on the team responsible for maintaining the OCA operating system, explained the rationale of tracking the FreeBSD head branch.

First, Jonathan and his team find FreeBSD code to be generally very stable and high quality. Second, they prefer to quickly find and fix the relatively infrequent and mostly low-impact bugs they do encounter. Otherwise, Jonathan explains, a development team that waits for the long-term, or Stable, branch, may end up in what he calls a vicious cycle of infrequent merges, many conflicts/regressions, and ultimately slower feature velocity.

**"Running FreeBSD head lets us deliver large amounts of data to our users very efficiently, while maintaining a high velocity of feature development."**
**— Jonathan Looney, Netflix**

Tracking the head branch helps Netflix add features more quickly. They also find that tracking the head branch makes collaborating with others in the development community easier.

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

- FreeBSD
- NGINX
- BIRD internet routing daemon

## Throughput Efficiency

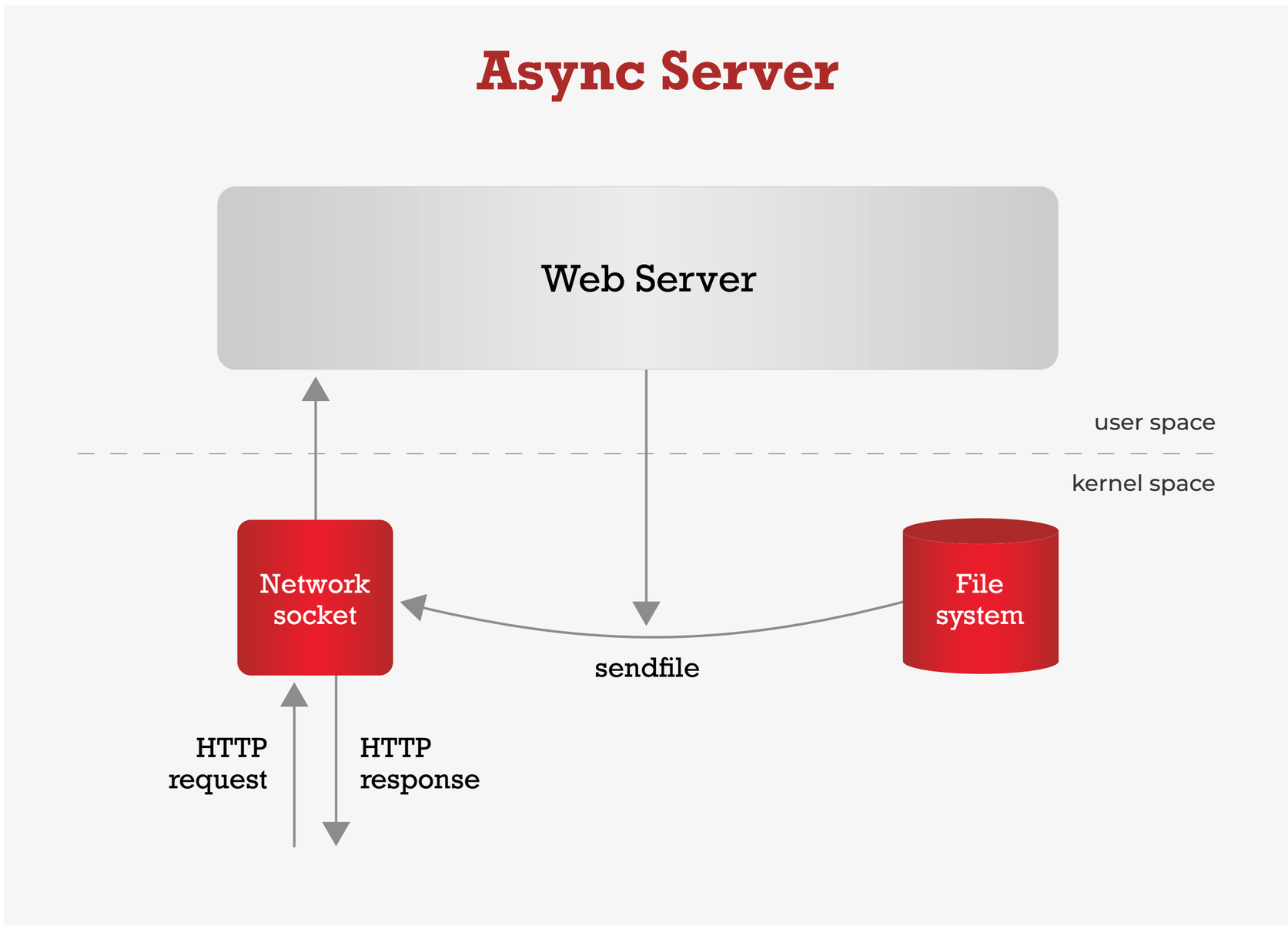
Just how efficient are these OCAs? Using FreeBSD and commodity parts, Netflix achieves 90 Gb/s serving TLS-encrypted connections with ~55% CPU on an Intel 6122 CPU, in 1 RU, with 96GB RAM, and 16TB of NVMe-attached flash storage.

Because it's their intention to upstream as much code as they can, all FreeBSD users benefit from the many enhancements that help Netflix achieve this kind of performance. Some of these contributions include NUMA enhancements, Asynchronous send file, Kernel TLS, Pbuf allocation enhancements, "Unmapped" mbufs, I/O scheduling, TCP algorithms, and TCP logging infrastructure.

In order to achieve this kind of performance cost-effectively, Netflix engineers realized they need to reduce context switching between Kernel and user space as much as possible. Async sendfile is one key technique that helps with this.

The new implementation of the sendfile(2) system call, which is a drop-in replacement for the previous one, speeds up TCP data transfers because it avoids copying file data into a buffer before it's sent. The new version of sendfile further speeds up and simplifies large data transfers by supporting asynchronous I/O.

The new sendfile is a product of a development partnership between NGINX and Netflix, and was released in tandem with a 2016 Netflix service expansion to nearly 200 countries.



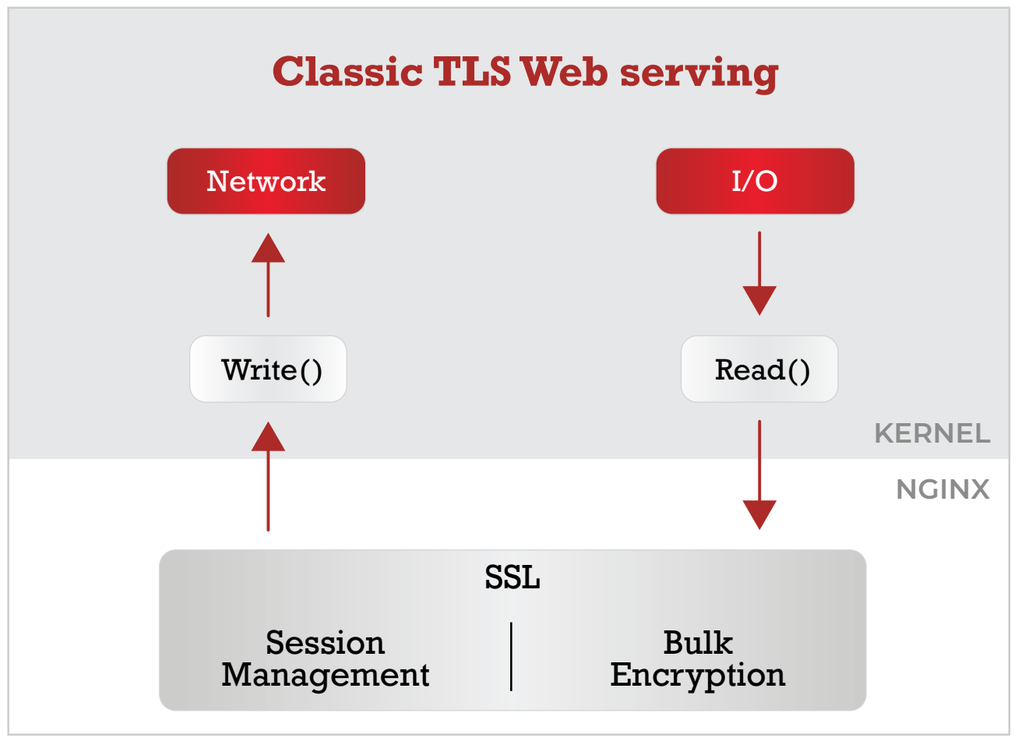

## Increasing Efficiency and Privacy — Kernel TLS

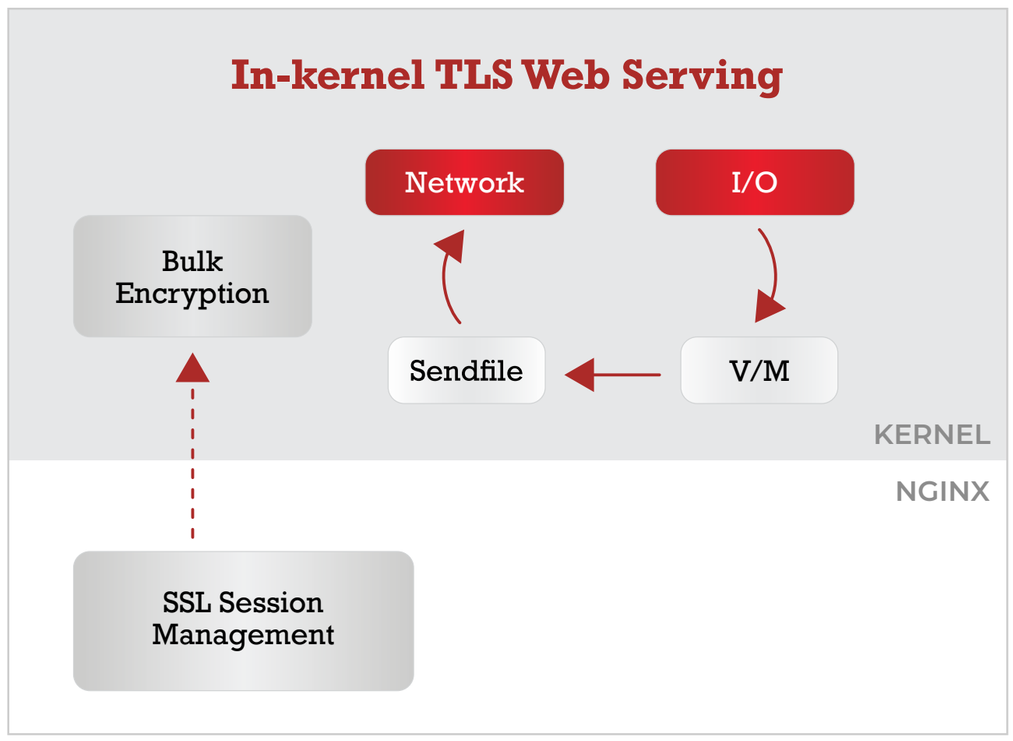
To protect the privacy of end users, in 2016 Netflix added Transport Layer Security (TLS). Jan Ozer summarized this move well in his Streaming Media article:

> Netflix had long deployed DRM to prevent piracy, and it protects customer data during account login and any administration via HTTPS. However, the actual transfer of the movie data was not protected, so any information contained in the communications between the server and client could be accessed by hackers, or by network administrators or ISPs. This information could be used to determine which content the viewer was watching, and perhaps other details.

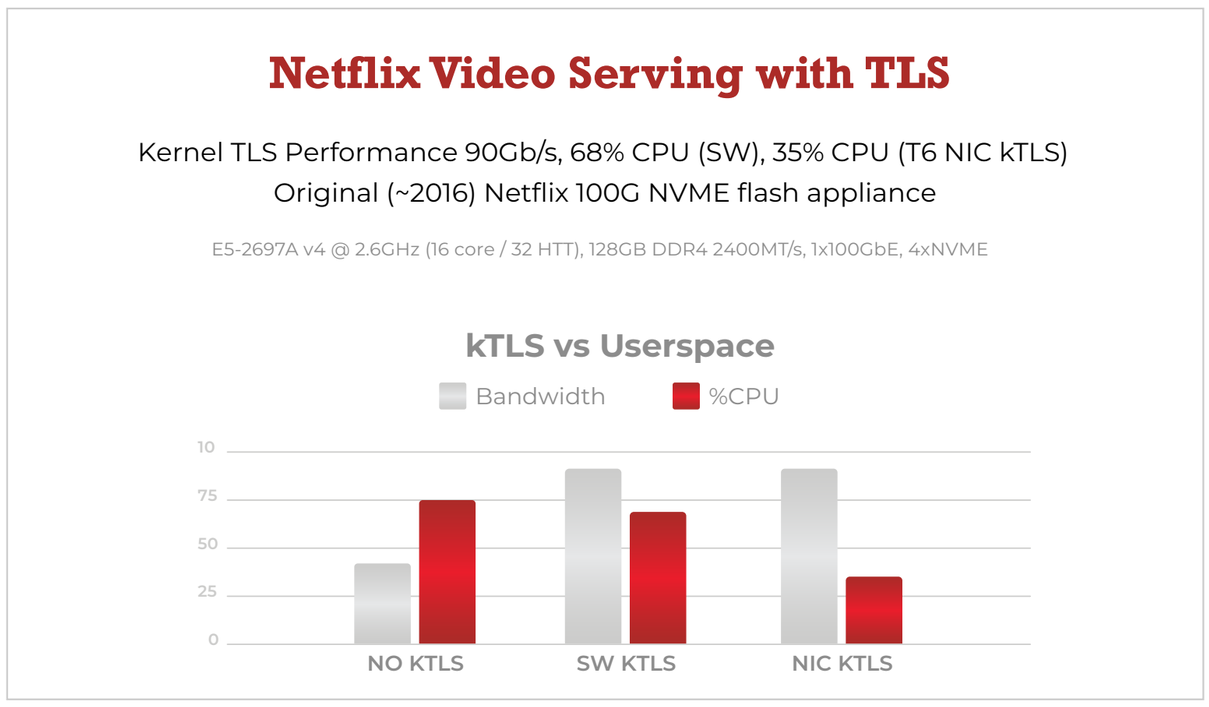
Adding TLS encryption *efficiently* required additional performance enhancements to the OCA software stack. That's because existing TLS techniques relied on the web server — an approach that Netflix's Director of Streaming Standards Mark Watson reported in 2014 would diminish capacity "between 30-53%."

The answer is kernel-side TLS, or kTLS for short, which marries TLS with the new sendfile model. This hybrid TLS scheme (described by John Baldwin in this vBSDCon 2019 session) keeps session management in the application space, and inserts the bulk encryption into the sendfile data pipeline in the kernel. TLS session negotiation and key exchange messages are passed from Nginx to the TLS library, and session state resides in the library's application space. Once the TLS session is set up and appropriate keys are generated and exchanged with the client, those keys become associated with the communication socket for the client and are shared into the kernel.




In their EuroBSD 2019 presentation, Drew Gallatin and Slava Shwartsman show how kTLS gives a 50 Gb/s boost to bandwidth performance while reducing CPU%. The next frontier in TLS performance improvement is something called NIC TLS, where the encryption is done in hardware. As the graph on below shows, this promises to reduce CPU utilization significantly.



## Getting to 200 Gb/s with NUMA

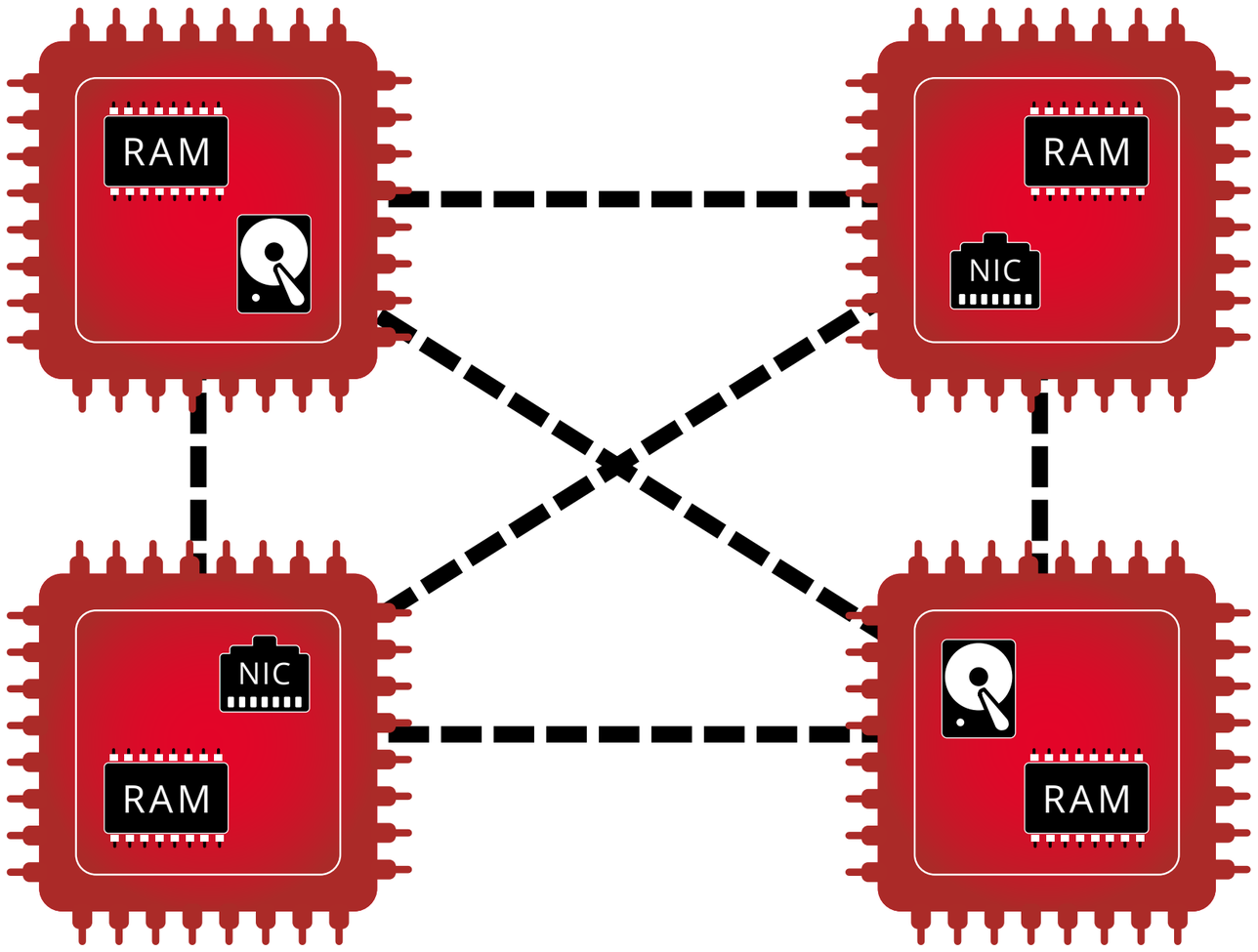
With no end in sight to members' demand for more shows and higher definition, Netflix continues to look for ways to increase the throughput of OCAs. With the evolution of high core count systems, the team has been developing and testing Non Uniform Memory Architecture, or NUMA, support since 2014, and that is now beginning to show results. Where a typical system has a single CPU, disk and memory, a NUMA system can have many more. As with sendfile and TLS, this can present throughput-sapping bottlenecks that Netflix engineers have been hard at work to minimize.

NUMA makes it cheaper for a CPU to access local resources (e.g. memory) and more expensive for it to access resources attached to another node. Consequently, memory and I/O locality impacts performance. For Netflix to take advantage of NUMA's greater computation density, they had to come up with a way to keep as much of the disk-to-CPU-to-network traffic local to a node and minimize performance-sapping NUMA bus transfers. This led to enhancements, which are in various stages of being merged upstream, including:

- Allocating NUMA local memory to back files sent via sendfile(9)
- Allocating NUMA local memory for Kernel TLS cryptobuffers
- Directing connections to TCP Pacers and kTLS workers bound to the local domain
- Directing incoming connections to Nginx workers bound to the local domain via modifications to SO_REUSEPORT_LB listen sockets

In tests, these enhancements have improved Xeon performance from 105Gb/s to 191Gb/s While reducing NUMA fabric utilization from 40% to 13%. For AMD EPYC, performance increased from 68Gb/s to 194Gb/s.

### Four Node Configurations are Common on AMD EPYC



## FreeBSD Gives NETFLIX Three Kinds Of Efficiency: Throughput, Development, and Operations

In response to the FAQ "why FreeBSD?" Jonathan says they came for the license and stayed for the efficiency — efficiency that Netflix measures in three ways:

1. Throughput, or performance, efficiency described in the previous section
2. Development efficiency
3. Operational efficiency

From a development perspective, the ease of working with the FreeBSD community helps Netflix upstream their enhancements for ongoing maintenance by the community. They also enjoy collaborating with others in the community that are working on the same area. Sharing code with these other community members can improve the code all parties are developing.

Finally, the huge fleet of OCAs requires sophisticated tooling for monitoring and operations. Some of the tools they've needed already existed, and the rest they have written. For the latter, Jonathan has found FreeBSD does a good job surfacing the necessary hooks and, where not, the team has been able to implement them.

## What's Coming Next from the Open Connect Brain Trust

In addition to NUMA and ongoing exploration of NIC TLS, the team is working on upstreaming some enhancements to kTLS and on UFS enhancements.

In closing, the massive scale of Open Connect combined with the team's focus on efficiency and their commitment to open source means that every FreeBSD user with a similar use case can reap the same performance benefits. The ability to turn on kTLS and take advantage of Async Sendfile allows anyone serving static content over HTTPS to extend their hardware lifetime, reduce density, and deliver a great user experience more efficiently.

---

**GREG WALLACE** is a freelance technology marketer who has been working with open source software and communities since 2005. In addition to his current work with the FreeBSD Foundation, Greg dabbles in Kubernetes, security, DevOps, and routing. Previously, he led marketing for Node.js, ODPi, and Hyperledger.