



Maintaining the World's Fastest Content Delivery Network at Netflix on FreeBSD

Netflix is a global entertainment company that revolutionized the way people consume TV shows and movies with its streaming service. Headquartered in Los Gatos, California, Netflix has grown into one of the world's leading streaming platforms, boasting millions of subscribers in over 190 countries. Known for its extensive catalog of films, television series, and documentaries, including critically acclaimed original productions, Netflix continues to shape the entertainment industry by investing in innovative content and technology.

CHALLENGE

With the escalating demand for high-quality streaming services, Netflix was tasked with sourcing an operating system capable of efficiently handling massive global traffic volumes. The system needed to be stable, scalable, and high-performance, yet flexible enough for extensive customization and optimization to meet Netflix's specific operational requirements.

SOLUTION

Netflix adopted FreeBSD-CURRENT for its Open Connect CDN, aligning closely with the FreeBSD community and staying at the forefront of system development. This strategic decision allowed Netflix to develop and maintain a customized FreeBSD implementation, enabling deep system-level optimizations and continuous enhancements tailored to their CDN's needs.

IMPACT

Netflix's strategic use of FreeBSD-CURRENT has significantly enhanced system performance and scalability. Through key contributions, like optimized **sendfile** operations and advanced transport layer security (TLS) support, Netflix has effectively managed extensive internet traffic, demonstrating the formidable power of open source collaboration in high-demand, large-scale environments.

The fastest and highest-trafficked network on the internet, all running FreeBSD

Gleb Smirnoff is a skilled software engineer and experienced FreeBSD committer who works at Netflix and manages the customized and performance-optimized FreeBSD-based firmware for Open Connect, the company's content delivery network (CDN).

During his presentation at the FreeBSD Vendor Summit in November 2023, Smirnoff emphasized the massive scale of Netflix's operations.

"We are one of the biggest sources of traffic on the internet – sending terabits per second, powered by thousands of servers or appliances, all running FreeBSD."

As Smirnoff notes, Netflix's Open Connect originally operated on a standard FreeBSD platform, which was gradually improved for better performance. In 2012, a proof-of-concept CDN was started on vanilla FreeBSD 9.0-RELEASE and nginx that was provisioned on servers equipped with a single 10 Gbit/s interface.

Over time, it became evident that achieving rapid growth required exceeding the limits of the operating system's current capabilities. The expected scale of Netflix's Content Delivery Network (CDN) was so massive that it was worthwhile to invest in the ongoing open source development of FreeBSD.

Netflix realized that when deploying a CDN at a global scale, even a single percentage point increase in performance results in savings worth hundreds of thousands of dollars. Netflix's customized version of FreeBSD enabled deeper integration and more precise optimization at the kernel level, leading to significant performance improvements.

Tracking FreeBSD-CURRENT at Netflix

Netflix carefully balanced its modifications with the need to stay aligned with the FreeBSD project's core codebase. This ensured that their custom enhancements improved the system's capabilities without causing an unsustainable divergence from the original FreeBSD source. This delicate balance allowed Netflix to leverage FreeBSD's strengths while creating a tailored solution that met their specific high-performance needs.

Another Open Connect team member, Drew Gallatin, detailed insights into FreeBSD's customization at Netflix during his November 2023 talk at OpenFest Bulgaria, a prominent technology and open source conference.

With over 25 years of experience contributing to FreeBSD, Gallatin shared his journey and challenges in optimizing FreeBSD for Netflix's Open Connect and emphasized the strategic decision-making process behind tracking FreeBSD-CURRENT, stating:

"We decided what we were doing was silly, and what we should do is track FreeBSD-CURRENT. It sounds crazy because that's where everybody pushes all their stuff, but it's actually the best thing in the world for us."

During his talk, he also shared anecdotes from the "Magical Mystery Merge," illustrating the importance of running the CURRENT branch. Gallatin explained, reflecting Netflix's proactive approach to maintaining system performance and stability:

"When we run FreeBSD-CURRENT, we catch things really fast. If there's some regression, we catch it right away. There's no two—or three-year delay between somebody committing something and us finding it's a problem."

Adding to the narrative on the subtree integration, Gallatin pointed out the benefits of this approach, highlighting the streamlined development and maintenance processes that resulted from Netflix's strategic alignment with FreeBSD-CURRENT:

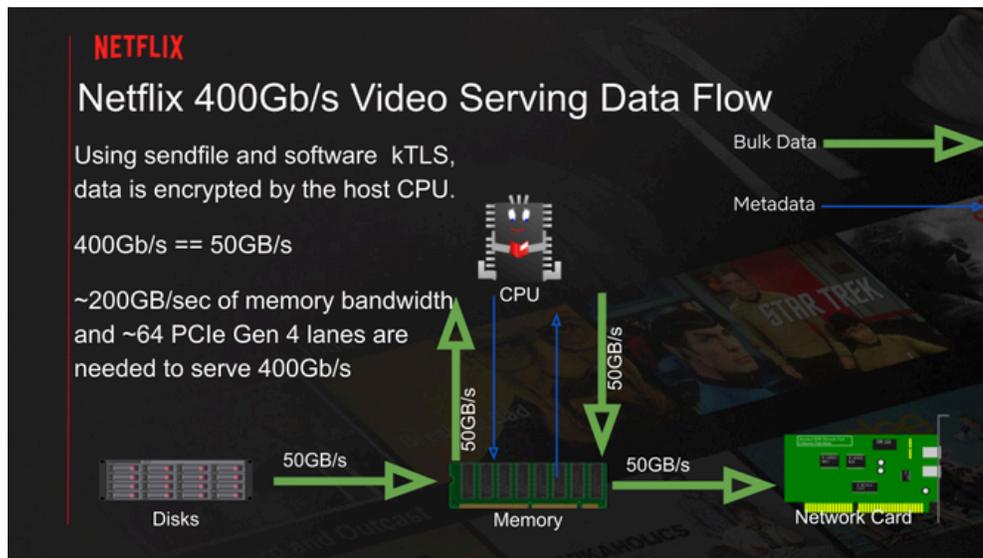
"Our tree is almost identical to the upstream FreeBSD tree... it greatly reduces the technical debt we accumulate by keeping our own patches."

Strategic integration and performance optimization of FreeBSD

Netflix carefully manages the code flow between the in-house FreeBSD implementation and the wider FreeBSD community. A rigorous testing framework, continuous integration, and unit testing are the foundation of Netflix's development strategy. Regular merges include upstream changes, and special focus is given to incorporating performance-enhancing patches ahead of their official inclusion in FreeBSD. A/B testing is performed for each merge to maintain or improve system performance and stability.

The evolution of Netflix's FreeBSD implementation involved refining the kernel to alleviate performance bottlenecks and handle the increasing data traffic, which includes RACK (Recent ACKnowledgment), a TCP stack developed by Randall Stewart, designed to improve the performance and reliability of data transmission. Other notable enhancements to FreeBSD by Netflix include asynchronous sendfile operations, which facilitate non-blocking data transfers, and advanced VM page caching techniques that improve data handling efficiency and network throughput.

The Netflix CDN team has also notably collaborated with the FreeBSD community to enhance the security and efficiency of data transmissions using Kernel TLS (KTLS).



KTLS is a technology that moves the processing of TLS (Transport Layer Security) from user applications to the operating system kernel. This improves performance for file and web servers using sendfile(9) by encrypting the data in the kernel, where it resides, and avoiding extra copying of the data into and out of user-space just to encrypt it. KTLS is helpful for high-throughput applications, like web servers, that require secure data transmission. It allows for efficient data handling and has enabled Netflix to achieve 400 Gb/s throughput on its CDN servers. Gallatin explains:

"We had the first 100 gigabit per second production CDN server in the world... due to Kernel TLS."

"What's kernel TLS? We moved bulk crypto into the kernel (from nginx) to preserve the sendfile pipeline."

"With sendfile and kernel TLS, we can eliminate many of these memory bandwidth bottlenecks, and now things become much more possible. By accounting for bandwidth and CPU utilization, we get about 375 gigs at about 53% busy CPU with FreeBSD."

Kernel TLS in FreeBSD is a large project and has undergone significant development through collaboration within the community. While at Netflix, Scott Long first proposed integrating TLS into the kernel. Together with Randall Stewart, they developed the foundational software TLS transmission mechanisms. Drew Gallatin contributed significantly to the project by introducing external pages mbufs and M_NOTREADY mbufs, which were essential for handling encrypted data within the kernel. He also developed a pluggable interface for various software TLS backends.

Later versions of KTLS made notable enhancements to the system. For instance, for FreeBSD 13, the transmission of Transport Layer Security (TLS) through offloading to network interface cards (NICs) was added. Drew Gallatin first implemented this feature in collaboration with Chelsio, which co-sponsored the project with Netflix for Chelsio T6 adapters. Later, [Hans Petter Selasky](#) extended this functionality to include Mellanox ConnectX-6 Dx adapters, enabling support for a wider range of hardware acceleration.

This ongoing development, backed by contributions from Netflix, Chelsio, and Mellanox, highlights the strong, community-driven efforts to enhance FreeBSD's network security and performance capabilities.

Giving back to the community

Netflix's strategy in managing its FreeBSD implementation for Open Connect reflects a deep commitment to the broader FreeBSD community. Smirnoff highlighted the significance of aligning closely with FreeBSD's development:

"It's crucial to reduce the divergence of your operating system to FreeBSD, which means that you need to upstream your changes."

He also articulated the practical benefits of this strategy, explaining,

"Tracking FreeBSD-CURRENT ... allowed us to collaborate with upstream developers and get our changes into FreeBSD quickly."

This approach has minimized technical debt and facilitated rapid incorporation of the latest features and improvements, keeping Netflix at the forefront of technological innovation in streaming.

Lessons learned and best practices

The successful management of large-scale FreeBSD implementations, such as Netflix's, provides valuable lessons on the importance of community involvement and open source collaboration.

- Engaging with the community early on and proactively contributing to the project is crucial to harnessing FreeBSD's full potential. These practices ensure that any adaptations to the system align well with ongoing developments in the broader ecosystem.
- Over time, refining the strategy for managing an organization's FreeBSD implementation by prioritizing community engagement, regular testing, and strategic upstream contributions can yield significant benefits.
- Adopting new FreeBSD features and conducting thorough testing to identify potential system degradations early in development is essential. This proactive approach helps maintain a clear understanding of how an organization's customized fork diverges from the primary FreeBSD Project, ensuring that enhancements improve the system's capabilities without leading to unsustainable divergences.
- Having a well-defined process for integrating external code and managing internal changes is critical. Setting clear protocols for code review, integration, and testing is vital to maintaining system integrity and performance.

By adopting these practices, organizations can effectively manage their FreeBSD-based systems, ensuring they meet specific operational needs while staying ahead of technological advancements.

Future directions

Netflix is committed to using FreeBSD's flexibility and performance capabilities and will continue collaborating with the community, focusing on growth and innovation. Netflix has set a precedent in the industry by successfully maintaining a customized FreeBSD implementation through strategic foresight, rigorous testing, and active community engagement.

Getting started with FreeBSD

Reflecting on Netflix's journey with FreeBSD, Netflix's CDN team offers valuable advice to organizations considering using FreeBSD. They suggest proactively engaging with the FreeBSD community and leveraging resources like The FreeBSD Foundation, which can provide crucial support on technical issues, implementation challenges, and community connections. For Netflix, the strategy was not just about adopting FreeBSD but integrating it into its ecosystem, contributing to its development, and sharing its innovations upstream.

The FreeBSD Foundation can assist with technical and implementation questions, networking, and connecting community members. If your organization is thinking about getting started with FreeBSD, email the Foundation using the [Contact Us](#) page of their website, or [download FreeBSD](#) to get started today.