
VARNISH PLUS+
VARNISH SOFTWARE
White Paper
HTTP Streaming

Executive summary

HTTP streaming is here to stay – a fixture of the landscape, enabling both development/improvements and user demand. The ubiquity and inevitability of streaming poses challenges but even more opportunities – and is a requirement for a wide array of media and non-media industries alike.

Varnish Cache and Varnish Plus form a solution that addresses the future-facing challenges of HTTP streaming delivery management and build on its natural strengths: performance and flexibility, both of which are major concerns as the future of streaming unfolds.

Introduction

Multimedia streaming has matured alongside the development and growth of the internet and faster, more widely available networks through which to access larger streams of content. The history of streaming highlights numerous attempts to make it into a widespread practice; a number of early hit-or-miss solutions showed the promise of the internet as a platform for delivering multimedia content, but took some time to deliver on the potential.

Eventually we arrived at the current state of streaming, which is almost ubiquitous across devices and includes both free and paid streaming services, ranging from on-demand YouTube videos, TV/media outlets streaming content globally 24/7 and their VOD offshoots, radio stations and premium content streaming from Spotify to Netflix and competitive contemporaries as well as enabling everything from online conference calls/meetings, webcasts and live events and e-learning/distance education. The potential for mass streaming was always there, but a variety of technologies had to converge to make streaming both scalable and seamless for the end user.

Varnish Cache and Varnish Plus can be used to bring value and high-performance to streaming media delivery; this paper will explain how.

From a trickle to a stream: Live and on-demand HTTP streaming

Early streaming* in the mid-1990s held promise but was not an easy road, with competing, proprietary technologies often working at odds to create an experience that was neither smooth nor user friendly. Until finally the different approaches looked to HTTP, the universal transport.

Riding the HTTP rapids - the universal transport
Enter HTTP streaming.

By far the most successful application on the internet – the web – relies on the HTTP protocol for transport. It follows that an overwhelming majority of internet-connected users will support content transported over HTTP. Other protocols often run into problems crossing firewalls and routers.

This, combined with its simplicity and the extensive tooling, has made HTTP into a generic protocol that has found uses far outside its original designation. Given its universality, HTTP has been instrumental in the evolution of video streaming. As with the entire history of media streaming, it has been an iterative process.

Gentle HTTP waves

The growing demand for streaming media led to the introduction of HTTP-based adaptive streaming. Streaming of the past, a mix of different and incompatible protocols, was not keeping pace. The near-ubiquitous HTTP protocol could be used to send media files in chunks, interacting with the streaming media player application to gain insight into network conditions. That is, only sending appropriately sized file chunks to suit the available network bandwidth, i.e. adaptive streaming. Some of the fundamental roadblocks to mass streaming were averted: endless buffering and connectivity problems could be circumvented using content distribution over standard HTTP (via content delivery networks) and caching.

This is not the end of the story and only illustrates how some of the platform-agnostic building blocks came to form the foundation of what we know as streaming today. There were still growing pains, including numerous shortcomings in terms of efficiency and speed.

To rectify these shortcomings, the chorus of competing HTTP-based transport layers emerged, such as, Microsoft (Smooth Streaming), Apple (HTTP Live Streaming or HLS) and Adobe (HTTP Dynamic Streaming or HDS), all of which are based on the simple principle of splitting H.264 video up into short segments and sending each of these in a HTTP response. To avoid the pitfalls of competitive infighting and incompatible protocols, the aim of interoperability has driven development.

This is essentially where we are with streaming today. HTTP-based adaptive streaming has become the de facto standard and has contributed to fundamental changes in several industries: entertainment and traditional broadcast (high-profile global events, such as the Olympics or FIFA World Cup, are jewel-in-the-crown streaming showcases that lend themselves to integrated digital campaigns; broadcast television has shifted both in the sense that it must offer live-streaming and on-demand services to compete with premium on-demand services, such as Netflix or Amazon Prime's instant video service). This is just the beginning, though. The reach and exponential growth of streaming (linear video, VOD or audio) – and its integration in virtually everything (from entertainment to higher education to marketing and so forth) – means that not only is it here to stay, it will continue to change and mature to follow use patterns.

Streaming Challenges

Apart from dealing with manageable challenges, such as overcoming legacy streaming problems and enabling cross-device, cross-platform delivery, the real challenges for streaming revolve around quality and volume. With more and more content and its ubiquity, users demand high-quality, fast, seamless delivery and complete, integrated digital experiences. Network bandwidth limitations still hold back quality improvements. To some degree this challenge can be met with developments to video coding and compression efficiency. The next generation of high-efficiency video coding (HEVC) in the form of H.265 (which will supposedly need only half the bit rate of H.264) can tackle some of the near-term challenges[1] and is already here.

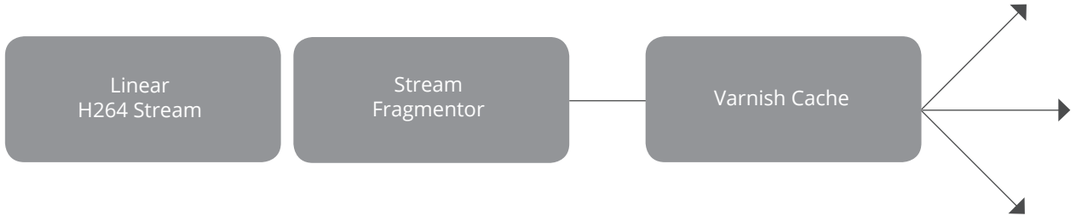
It may sound like a tired refrain but the present and future, as usual, are defined by performance, both the quality of the performance itself and the quality of the end-user experience. With more multimedia web content requiring high-performance and reliable delivery across devices, continuous development is essential.

---

1 http://www.itu.int/net/pressoffice/press_releases/2013/01.aspx

Technology solutions: What Varnish offers

Streaming process with Varnish Cache



Bring on the flood
HTTP is what Varnish Cache and Varnish Plus (the subscription-based, feature-packed version of Varnish Cache) are built for.

Varnish can be used as a standalone component for serving video as an efficient way to scale out your platform. It also enables serving massive amounts of content/traffic from a single location efficiently (high-volume VOD). Varnish is often deployed as an “origin shield” when used with CDNs. And, in fact, you can build your own DIY distribution network with Varnish as the caching engine. Flexible.

Features

Geo-blocking at country or city level

When content needs to be restricted by geography, there is a GeoIP VMOD for limiting or restricting access by specific location. While Varnish does not have this functionality built in, it is just one of the many flexible additional modules (VMODs) that can be added thanks to the flexibility of VCL. By setting a header indication instructing that a geo-based limitation should be put in place, we can set up specific restrictions.

```
sub vcl_recv {
    if (maxminddb.query_country(client.ip) ==
"FR") {
        return (synth(403, "Sorry, unauthorized
country"));
    }
```

Token access in any shape or form

Using VCL, it's easy to quickly prototype and test the logic of various authorization scheme. This is greatly helped by the digest vmod which provides a collection of hashing functions, enabling you to build and check HMAC very simply.

For example, here, we expect the client to hash the requested url together with ashared secret, and place it in the "check" header:

```
sub vcl_recv {
    if (req.http.check !=
        digest.hmac_sha256("secretkey",req.
url)) {
        return (synth(403, "Sorry, bad
secret"));
```

Flexible rate limiting and abuse suppression

For the more complex cases, it is also possible to create a VMOD to improve readability. Designed exactly for these use cases, the throttle VMOD will act as a guardian, directly in VCL, keeping tabs on requests, letting you refuse the too-frequent occurrences.

```
if (vsthrottle.is_denied(client.ip, 3, 1s)) {
    return (synth(429, "Too Many Requests");
}
```

Dynamic TCP buffer and latency management

You can limit a single client to download at an unfair rate. You can specify multiple limits to really fine-tune the access patterns. An example of this kind of use is, for example, when you have a VOD library available for streaming on mobile, and want to rate limit how much bandwidth is can be used.

Also note that because the first argument is a string, it's possible to filter IPs, as well as URLs, countries, and more, including combinations of those parameters.

Building on what open-source Varnish Cache offers, Varnish Plus adds even more:

- **Varnish Custom Statistics** (VCS) to provide real-time insights into how content is being consumed and by whom. By plugging itself into the super-fast shared memory log of Varnish, and being driven by VCL, VCS offers a versatile and powerful tool to profile data usage on your platform, in real-time and without slowing down delivery.
- **Varnish Massive Storage Engine** (MSE), which provides a significant amount of local cache storage. Caching live streaming is easy, thanks to the short lifetime of content, but to cache an entire catalogue of VOD, you may have to make compromises. But not with MSE, as it is designed to scale to tens and hundreds of terabytes, allowing to cache all you need and not just the hottest content.
    - Built for up to 100+ terabytes of storage on each node
    - Fragmentation-proof allocation algorithm
    - Higher cache hit rates due to LRU replaced with LFU
    - Optionally persistent datastore
- **Varnish High Availability** for content replication to reduce pressure on the origin and increase reliability of your service. Networks of CDNs turn to the origin to fetch content, and the pressure drives origin servers into the ground. Varnish can serve as a protective layer against the massive influx of traffic, acting as a tier of caches - horizontal scaling in front of the traffic.
- **Request coalescing:** Protects/reduces load on the origin server, which enables linear video
- **Access to Varnish core developers** who can help optimize implementation and configuration to ensure that your performance, setup and user experience are the best they can be.

What's in it for you? Varnish Plus benefits

With the core functionality drawn from the original open-source Varnish Cache heritage, Varnish Plus is constantly improving and developing, benefiting from feedback from our developers, from the community and from early-adopter enterprise customers.

Built for the now and the future
Built for high-traffic, high-volume, dynamic content, Varnish delivers scale, speed, performance and stability. Able to handle all kinds of files, all levels of traffic, Varnish helps you be ready for anything.

Scalability
Varnish helps to ensure effective content distribution and availability while managing peak traffic.

Flexibility
Easy and flexible configurability make Varnish highly adaptable to specific needs and conditions.

Secure delivery
Secure in both ensuring only the right audience can access (permissions) and secure in ensuring uptime, availability and speed of delivery.

Origin shield: Backend protection
Varnish can serve as an origin shield, protecting your backend servers from being overloaded.

User experience
Speedy, smooth responsiveness across devices is what users want and expect. Varnish helps you deliver what your users and customers demand.

Access to expertise
Access to Varnish core developers can help you optimize your implementation and configuration to ensure that performance and user experience are the best they can be.

Who should use Varnish solutions for streaming?

Varnish Cache and Varnish Plus for video streaming and distribution has been used by companies like Metacafe, SFR, Ericsson, Surfline, Dailymotion, and Globo, to name a few, as well as by partners who use Varnish as part of solutions they develop for their own customers.

Almost any company or entity may have a use for streaming – and scaling for streaming. We know live sporting events are driving innovation and incredible traffic, with the 2016 Rio Olympics forecast to be the most live-streamed event in history – with all future Olympics and other big events, such as the NFL SuperBowl in the US, which has struggled keeping up with traffic and delivering quality streaming[2] – steadily driving the numbers higher.

New uses, channels and users are being identified all the time (such as YouTube's recent drive to make 360-degree live streaming from devices possible and eventually the norm, or Facebook's Live for Facebook Mentions live video broadcast functionality). We also know that the move to cross-device and mobile streaming at ever-greater consumption levels (mobile is predicted to drive internet traffic past the zettabyte mark in 2016[3]) means that everyone will need to be vigilant about scale and optimization.

Everyone is cutting the cord in one way or another, which means that there are undoubtedly opportunities and use cases in multiple sectors and industries that we haven't even thought of yet. Now is the time to be creative – get ready for what's coming.

With these considerations in mind, virtually anyone could benefit from Varnish solutions for VOD and live streaming, including:

- CDNs
- Content providers
- Broadcast networks/channels/telecoms companies (who do not want to be seen only as the "dumb pipe" supplying the bandwidth)
- Media outlets – TV, radio, online
- Streaming content sites/aggregated content, such as Dailymotion
- Sports associations (as mentioned, the NFL has not managed on its own on this front very well, while Major League Baseball was so technically adept that they have their own spin-off, in-house streaming media department that handles not only baseball but streaming digital distribution for a lot of other very big media names[4]).
- Corporations, universities and other large institutions moving into live streaming of lectures and training, etc.

This is really only a sampling of who should be using Varnish for streaming.

---

2 http://www.itu.int/net/pressoffice/press_releases/2013/01.aspx#.VrFSiFLscqY
3 http://recode.net/2016/02/03/phones-will-drive-internet-traffic-past-the-zettabyte-mark-this-year/
4 http://www.theverge.com/2015/8/4/9090897/mlb-bam-live-streaming-internet-tv-nhl-hbo-now-espn

Get your feet wet:
Start HTTP streaming with Varnish

We have helped our customers globally to build advanced, scalable and fast streaming solutions on their own terms through the whole lifecycle of the software: Design, feature development and enhancements, implementation and optimization. Varnish Plus offers all the flexibility and performance of Varnish Cache as well as several modules that enhance the experience – in addition to expertise from Varnish core developers in tackling and resolving challenges and helping you get the most from your streaming opportunities.

*Appendix: History of streaming

Choppy waters: The early days of streaming
The 1990s held a lot of promise for streaming, but the stop-and-go, choppy nature of streaming content made it a frustrating enterprise. Nevertheless its origins date back to 1995, when the first live-streaming event – a Major League baseball game – was streamed live to subscribers.

Technological jump forward this may have been, but most enterprises and technologies occupying the landscape through the 1990s failed to capitalize on the promise, devolving into legal or technology-based skirmishes. Indeed, much of the technology supporting streaming at the time was short-sighted and proprietary (for example, Real Networks and Microsoft's Windows Media), which meant that widespread adoption and uptake never quite materialized. Connectivity at the time was also prohibitive – live streaming video over relatively slow networks would not inspire mass adoption.

It was only with the introduction of Flash that the future of streaming was born. For all its limitations, Flash was a jump forward on the technology side, emerging as the only cross-browser alternative for quite some time. Yet, network bandwidth continued to throttle the reach and scalability that would enable mass adoption by users and encourage consistent development from technology companies. But the potential of live streaming was finally clearly in view; now it would just be a matter of how.

Fording the stream: HTTP-based streaming
Like most developing technologies, a groundbreaking idea with truly industry, society and life-changing possibilities, often struggles to find its way. As the history of streaming media in the 1990s illustrates, development happened in fits and starts, usually within closed, proprietary systems that held back mass-scale adoption or use.

As the internet and the protocols governing it have matured, consumer expectations have demanded more. This has led to the "consumerization" of streaming multimedia, complete with stability, speed and high definition. Delivering on expectations has occurred in fits and starts, but once consumer demand fueled this progression, development has become progressively faster, with stability and scalability improving all the time. But what got us to where we are today? HTTP.

About Varnish Software

Varnish Software is a leader in the high growth Web Architecture Performance market. Using a flexible caching technology, our products and services help companies like New York Times, Vimeo, Nikon, Cachefly and Fujitsu deliver web and application content fast, reliably and at massive scale to provide exceptional customer experiences. Supported by an innovative developer community, our open source project Varnish Cache continues to flourish and makes 2.2 million websites worldwide run faster.

Varnish Software has offices in London, New York, Los Angeles, Stockholm, Oslo and Paris. For more information, please visit: http://www.varnish-software.com/

Notes:

VARNISH PLUS+

VARNISH SOFTWARE

New York +1 646 586 2052
Los Angeles +1 310-648-8474
Paris +33 607 47 36 90
London +44 20 7060 9955
Stockholm +46 8 410 909 30
Oslo +47 21 98 92 60