



Wowza Media Cache Deployment Architecture

www.wowza.com

Abstract

Wowza Streaming Engine™ software provides functionality to serve video on demand (VOD) content for large-scale deployments. The Media Cache feature in Wowza Streaming Engine is a read-through caching mechanism for VOD streaming. It lets you upload and manage content through a centralized point, allowing you to scale without having additional hardware, implement complex content management, tax bandwidth and network resources, or increase latency. This document describes a high-level architecture, system considerations, and network layout appropriate for large-scale deployment of VOD streaming that uses Media Cache and the HTTP functionality provided by it.

Wowza® Media Systems, the leader in media server software, provides reliable, scalable, and secure live and on-demand streaming. Wowza Streaming Engine software enables organizations of all sizes to engage millions of viewers worldwide by delivering the highest quality video and audio to any connected device.

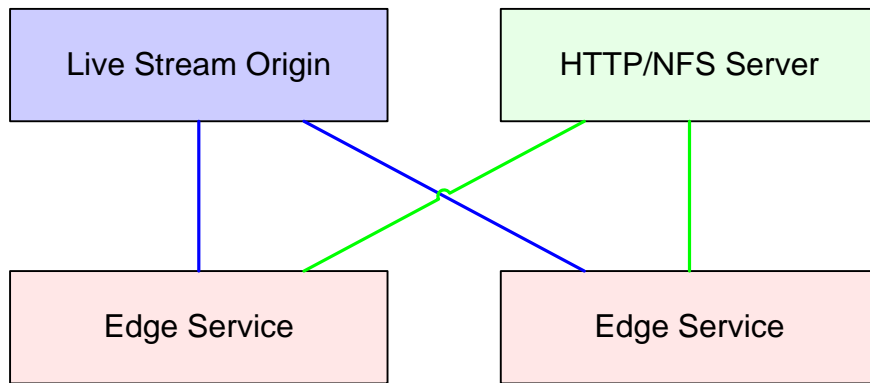
Introduction

Video content increases in resolution and file size, even as more distribution across public and private networks is required. Consequently, making media management as simple as possible becomes increasingly important. The Media Cache feature in Wowza Streaming Engine provides a simple and effective mechanism to achieve this goal in an extremely cost-effective manner.

This document describes a high-level architecture, system considerations, and a network layout appropriate for large-scale deployment of video on demand streaming using Media Cache and the HTTP functionality provided. It outlines considerations for network and content management, content access and distribution available within Media Cache, scaling of content stores, and local cache sizing. The final section of this paper presents some high-level deployment options.

High-Level Architecture

The high-level architecture for deploying a large-scale streaming service for both live and video on demand content is simple and has three distinct areas: Live Stream Origin, HTTP/NFS Server, and Edge servers.

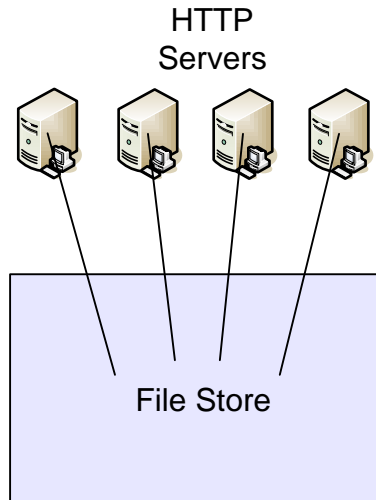


Any one of these functional areas can be expanded to provide resilience, scale, and high performance when using Wowza Streaming Engine software. This document focuses on how to use the Media Cache feature in Wowza Streaming Engine to scale video on demand streaming.

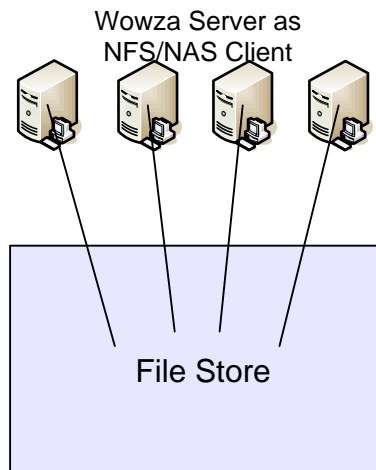
Media Cache Overview

Media Cache can be used to retrieve content via HTTP requests from either an HTTP server or from a local file location. The method of use doesn't alter how Media Cache manages content or how it makes content available to clients that connect.

When using HTTP functionality, Media Cache requires that the HTTP servers support range requests (which are part of the HTTP/1.1 specification) and have sufficient I/O to service requests from a significantly large number of edge servers. Extremely large deployments should provide adequate storage and performance to a cluster of web servers with access to the same file store or to multiple file stores that are mirrored. The following high-level diagram shows how HTTP servers can be used in front of a file store, which then allows Media Cache to access content.



The following high-level diagram shows how Media Cache can be used to access a file store directly.



The method used to access the file store depends on the specific deployment in question and how the file store is made available to Media Cache.

If a large number of clients will request different content at the same time, the disk storage required on the edge servers running Wowza Streaming Engine must be fast enough to service the diverse needs of all these clients. The disk storage also must be sized correctly to account for different asset sizes, how often assets are replaced, and how often assets are accessed.

A Wowza server can be tuned for specific environments to be more efficient with regard to content size. This customization can increase network efficiency and reduce impact on the HTTP store. When large disk arrays are deployed, onboard disk cache and OS-level caching make large reads possible and efficient.

For example:

```
File size in the file store - 2 GB
Default Media Cache Request size - 256000 bytes
Number of Media Cache access requests from Wowza Streaming Engine - 7812
```

If the default Request size is increased to 2 MB, then the number of requests from Wowza Streaming Engine for an asset drops from 7,812 to 1,000.

To get the best performance from Wowza Streaming Engine and the file store, it's essential to determine the average asset size when using this approach. The following is a simple calculation model that can be used to determine asset size:

```
HTTP Request size = Average asset size / 1000
```

It should be noted that the minimum HTTP Request size should be 256,000 bytes and the maximum should be 2,000,000 bytes to yield the best performance. The Request size can be configured via the **defaultBlockSize** property setting for the Media Cache Source in Wowza Streaming Engine Manager (**DefaultBlockSize** property within the **MediaCacheSource** definition).

To calculate edge storage size requirements (I/O speed is a different problem), make the following assumptions:

```
Average asset size is 450 MB (500 Kbps for 2 hours).
Average number of unique assets watched per day is 200.
```

The edge disk requirement to fit all assets for 1 day is 88 GB. Assuming that growth occurs at 5 percent per month over 18 months, then 212 GB of edge storage should be deployed.

To ensure adequate I/O and seek speed, deploy a set of RAID 1 or RAID 5 disks (depending on the operational requirements). Both of these disk configurations offer similar speeds and redundancy (parity). The number of RAID disks to use depends on the number of Wowza Streaming Engine servers that are deployed in a location.

Note

When the assigned cache space in Media Cache is full and no expiration can be performed, Wowza Streaming Engine tries to get assets from the defined source as they're requested. This causes performance to degrade. Wowza Streaming Engine does this on a per-block basis, and this can cause a significant increase in the number of HTTP/File access requests, depending on the number of assets being requested that don't fit.

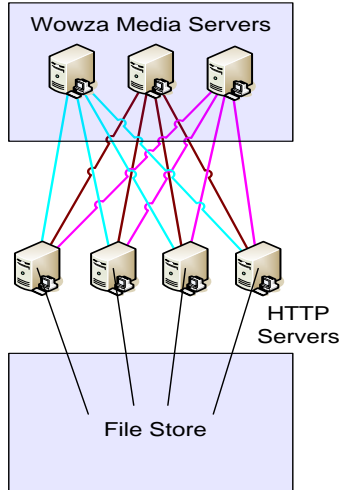
When using Media Cache, the I/O speed of the cache space is highly important. For content greater than or equal to 212 GB, using RAID or solid-state disk (SSD) sets is essential. Wowza Streaming Engine performance metrics show that 2 x 36 GB, serial-attached SCSI (SAS), 15K RPM hard drives with RAID 0 (striping) enable 700 concurrent connections to 400 files. In this configuration, each file will stream at 512 Kbps with no data loss. This metric can be applied to Media Cache because assets that are downloaded from the file store are essentially accessed like any other VOD asset.

When deploying a Media Cache, cache location is important to balance read and write rates. In normal operations, the number of read operations exceeds the number of write operations; however, it's very important to determine the write rate that will be available and how sustainable this rate will be. A good rule to follow is to assume that 10 percent of the performance of the cache location will require writes. If the overall throughput of the storage location can be determined—200Mbps, for example—then the write rate should be set to 20Mbps. Setting the write rate too low will cause either a significant bottleneck to the Media Cache system or any retrieved assets will be corrupted when they're written. The write rate can be configured via the **Write Rate** setting for the Media Cache Store in Wowza Streaming Engine Manager (**WriteRate** property within the **MediaCacheStore** definition).

Example Wowza Streaming Engine Deployment

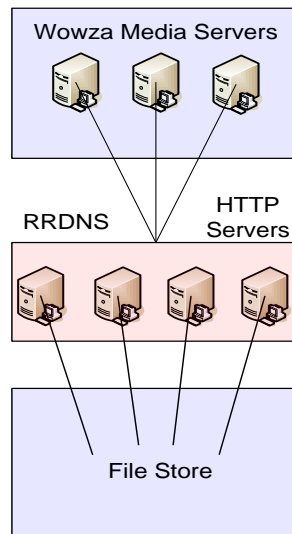
Wowza Streaming Engine can be configured to use multiple HTTP servers for the same content in a load-balancing manner or for different content. The most common deployment method is to load-balance across multiple HTTP servers when multiple requests are received for the same content, as shown in the following high-level diagram.

WOWZA MEDIA CACHE DEPLOYMENT ARCHITECTURE



Media Cache can be configured for HTTP timeouts, so if one HTTP server fails, service should continue, though initial requests for content may take a little longer during the timeout period.

An alternative deployment strategy is to use Round-robin DNS (RRDNS) or Geographic DNS for multiple HTTP servers. Operationally, using one of these alternatives has some advantages. If a failure occurs, a change in DNS can resolve the issue without the need to change any Wowza server configurations.

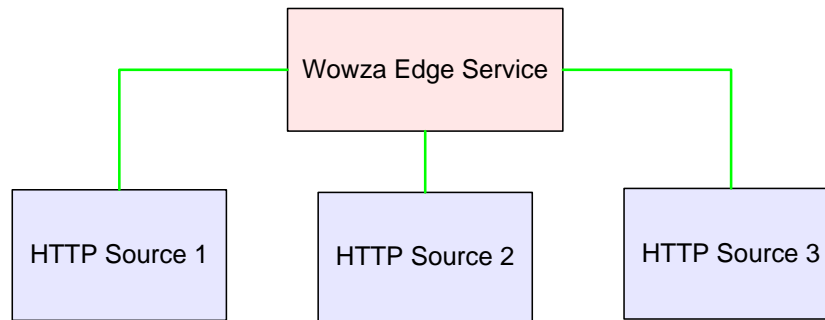


Changes to the Media Cache configuration require that the Wowza Streaming Engine be restarted, so this should be considered when deploying the service.

Multi-Location HTTP Servers

When using Media Cache, it's possible to deploy multiple HTTP source servers and have them referenced independently in the content request that's received by Wowza Streaming Engine. This is a configurable aliasing mechanism so you can partition by different customers or by content type (movies, songs, advertising, and so on).

The following diagram shows a Wowza Streaming Engine that connects to three independent HTTP servers for different content. The content for each of these servers is cached in the same way as in any other Media Cache deployment, so sizing must be considered.



These different HTTP sources would be referenced in any request made to Wowza Streaming Engine. The following examples show how they could be accessed:

```
rtmp://[wowza-ip-address]/vod/mp4:http1/mycustomerfile.mp4
rtmp://[wowza-ip-address]/vod/mp4:http2/mytypefile.mp4
rtmp://[wowza-ip-address]/vod/mp4:http3/myscalefile.mp4
```

The aliases shown above (http1, http2, and http3) would be defined in the Media Cache configuration and could point to a specific HTTP server.

Summary

Media Cache technology in Wowza Streaming Engine provides a simple and effective method to scale an on-demand service. It's important to understand the Media Cache options and how they relate to any content that's deployed.

Points that should be considered when deploying a Media Cache system include:

- Centralized network capacity
- Centralized storage capacity
- Storage growth rate
- Average asset size
- Asset churn over a 30-day period
- Edge cache capacity and I/O speed
- Specific write rate

References

[Wowza Streaming Engine Media Cache User's Guide \(PDF\)](#)

[How to scale video on demand streaming with Media Cache](#)