Media Services On Demand: Encoder Best Practices

March 03, 2022

Contents

Overview

Welcome to Akamai Media Services On Demand: Encoding Best Practices.

Media Services On Demand is a cloud-based technology platform for live and on-demand streaming. It is proven to support the largest online events by managing millions of simultaneous streams in a single event and delivering true 1080p on-demand premium content. Media Services On Demand is a platform for the next generation of streaming that is designed to address problems of the last generation by, for example:

- Simplifying live and on-demand streaming to multiple devices and run times.
- Delivering an interactive television-like experience that complements traditional television.
- Protecting content with powerful, responsive, and easy-to-provision security.
- Raising the bar on scale, performance, and reliability.
- This document provides best practices for creating high-quality video-on-demand experiences using Media Services On Demand. It explains how to find information about creating the best source material, selecting the best transcoding options, and ensuring the content is prepared for delivery. This document provides generalized best practices. Each customer's situation is different. Contact your Account Representative for assistance with your specific needs. This document contains the following sections:
- *Source preparation best practices* on page 3 provides recommendations and describes the following:
  - *Source preparation best practices* on page 3
  - *Transcoding best practices* on page 5
  - *Delivery best practices* on page 9
- *Encoding recommendations* on page 11 provides recommendations on selecting bitrates.

Source preparation best practices

The following recommendations are for producing the original source material, which you later transcode into multiple versions for delivery:

Basic characteristics

To ensure the highest quality playback experience, you should always begin with the highest quality content possible. This is a standing rule for video transcoding and applies to the entire workflow from capture to delivery.

However, there is a trade-off:

While better source will always create better output, there is a point at which the additional value of working with a higher quality source diminishes and is simply a waste of storage, transit, and time. Assuming your entire production workflow is of the highest quality, the source to use for transcoding can be as small as twice the bitrate of your highest possible deliverable derivative. For example, if you plan to target a maximum bitrate of 8 Mbps, 1080p derivative, you should aim to produce your source content at 16-25 Mbps using H.264. You should consider a higher bitrate if using less efficient video codecs. Transcoding from a source that is higher quality than this is unlikely to provide any noticeable quality improvement.

While working with a source of higher quality is certainly acceptable, it requires additional storage and processing time, and might be difficult to move from location to location. For example, every hour of footage encoded at 10 Mbps consumes roughly 4 GB of storage. A two-hour feature film at 100 Mbps consumes roughly 90 GB of storage and might not produce much visible improvement over the same film encoded at 20 or 30 Mbps when viewed on a desktop or mobile device.

Source content can be in any number of different container or codec combinations, but you should always try to work with source content that is lightly compressed in an industry-standard format to ensure compatibility and quality. Using MPEG containers with the H.264 codecs is highly recommended.

Tracks

At most, source material should always have a single video track and a single stereo audio track. Including multiple video and audio tracks might produce unexpected results, as only the first tracks are used. Content containing additional tracks for metadata, such as subtitles, can be included and passed through to created derivatives.

Interlaced content

Interlacing is often found on content originally created for display on a television, as opposed to a digital device. This type of footage is created by running “half frames” at twice the frame rate by drawing every other line and then filling in the remaining lines on a second pass. On a digital screen, both frames must be combined and displayed at the same time. This results in noticeable lines in footage that are particularly bad when there is motion in the video.

There are several methods available to de-interlace content, each with their own benefits, but the recommendation is to correct the interlaced footage as early in the production process as possible to ensure the highest quality. Due to how the de-interlacing process works, it is very important that it is done before applying additional modifications such as frame scaling. Attempting to de-interlace footage that has been modified from its original state produces noticeably bad results.

Aspect ratio

To ensure video content is not stretched inappropriately during playback you should ensure your source content is encoded with the correct display aspect ratio and square pixels.

Ensuring your aspect ratio is the same as your playback locations will prevent black bars from being displayed on your content during playback. While different devices might have different aspect ratios, you should try to make the content match as closely as possible.

In addition to display aspect ratio, you must also ensure your content has a 1-to-1 pixel aspect ratio — your pixels must be as tall as they are wide. This is another problem that is uncommon with completely digital footage but is often found with content that was transferred from older mediums such as DVDs, film, or anything using the PAL format that is common in Europe.

Because most devices are unaware of the pixel aspect ratio of played content, they simply display them as square. This means your content is displayed skinnier than appropriate so things will appear taller and skinnier than they did in the original footage.

While this issue can be corrected in the transcoding process, it is best to work with source material that has been corrected as early in the production process as possible.

Transcoding best practices

Consider these recommendations for the following when producing the actual derivative videos that will be delivered to the client:

Keyframes, GOP size, and alignment

Keyframe interval, also known as keyframe rate or GOP (group of pictures) size, should be selected based on how the content is to be delivered. To ensure compatibility with multiple delivery protocols, we suggest you use keyframes every two seconds. This ensures that they fit evenly into delivered segments for streaming protocols that are generally six or ten seconds long. Using alternative keyframe rates might be warranted based on the content type (such as, slow-motion talking heads or high-motion sports). In these situations it is worth evaluating your keyframe rate in conjunction with the selected delivery protocols to ensure compatibility. In addition, all keyframes must be of type IDR to ensure frames in one segment do not require data from another segment to display properly.

Keyframe interval must be identical across different renditions of the same video to allow mid-stream switching, allowing the best possible renditions to be delivered at any given moment

Some encoders have an option to enable scene change detection that allows keyframes to be inserted when a scene change occurs. This improves visual quality by allowing the entire frame to be redrawn when necessary. Due to the extra keyframes possible with this setting, you can raise the overall bitrate of the video.

Keep in mind that adaptive bitrate streaming still requires consistent keyframes within each rendition and is shared across the rendition set. Therefore, you should only enable scene change detection when it can be in conjunction with a minimum keyframe internal option. This enables the encoder to insert keyframes even when a scene change is not detected.

Frame rate

The frame rate of different renditions should be kept the same across all renditions that any one device could play back. Whenever possible, ensure that your renditions have the same frame rate as your source, as changing frame rates will have a noticeable negative quality impact.

Shared attributes of renditions

To ensure high quality playback and efficient switching between renditions during playback, all renditions for a given video must share the following:

- GOP size or keyframe rate
- GOP alignment
- Frame rate
- Audio sample rate and bitrate

Reference frames

A reference frame can be used by a P-frame, also known as a predictive or partial frame, to help define a future frame in a compressed video. While H.264 can support any number of reference frames, we recommend using between three and five (three to five reference frames can be used by a P-frame) for optimal performance across devices.

H.264 levels limit the decoded picture buffer. For example, if level 4.1 is being used, then the maximum reference frames allowed for 720p and 1080p videos are nine and four, respectively. It is important to keep the number of reference frames below the allowed H.264 level limits of the targeted device. For more details, you can view the full H.264 profile and level limitations table here:

B-frames

B-frames are a type of partial frame in compressed videos that are generated by referencing data from previous and future frames. While this can dramatically improve the efficiency in terms of compression ratio, they do require additional CPU. We recommend using three B-frames per GOP for devices that support B-frames. For videos that contain animated material, you can safely use a higher number of B-frames per GOP.

Some encoding tools allow you to use adaptive B-frames, meaning the encoder can place B-frames arbitrarily if it determines it will improve visual quality. This might improve quality on devices that support B-frames. Only devices that support H.264 Main profile and above support B-frames.

Many encoding solutions have the option of using pyramidal B-frames, which allow a B-frame to reference another B-frame for visual data. Unfortunately, support for this across devices is not complete and can cause unusual playback anomalies such as looping footage. This type of B-frame is specifically not supported by Apple Inc. HTTP Live Streaming that currently is used by a significant portion of mobile viewers, so it should be avoided, unless you do not want to reach mobile devices.

Encoding entropy algorithm

There are two coding options available for generating H.264 content:

- Context-Adaptive Binary Arithmetic Coding (CABAC)
- Context-Adaptive Variable-Length Coding (CAVLC)

We recommend using **CABAC** encoding as it provides a 7-10% quality improvement over **CAVLC**. However, because it is more efficient, it requires an extra 10-15% CPU. **CABAC** encoding is only available in H.264 Profiles Main & High. When targeting low-powered devices, such as older cell phones and tablets, we recommend using H.264's Baseline Profile that uses the less compute intensive **CAVLC** encoding entropy algorithm.

Weighted Prediction is another encoding option available in H.264 that can improve quality by improving motion compensation. It should be avoided, though, on less powerful clients due to the extra CPU power it requires and its limited compatibility in the lower codec profiles.

Rate control

During the encoding process you have the option of encoding your renditions with a constant bitrate (CBR) or variable bitrate (VBR). Variable bitrate can produce higher quality because it allows the encoder to use varying amounts of data to represent different sections of the video (a complex scene of action would have more data than a simple scene of a talking head). Unfortunately, due to the limitations of bandwidth and CPU across different clients, VBR might cause playback problems if the data rate suddenly jumps to a rate with which the client can not keep up.

Because of this, it is recommended to use CBR as it maintains a steady stream of data regardless of the complexity of the video in any given frame. This ensures client playback is uninterrupted. Advanced users should consider using VBR if their encoder allows them to cap the data rate used by the encoder to 1.5 times the desired bitrate. This allows some flexibility and should prevent massive jumps that could interrupt playback.

Regardless of the method used it is highly recommended to use two-pass encoding whenever possible. Two-pass encoding performs an analysis step before the actual encoding step, which allows the encoder to effectively distribute data rates and align keyframes.

While it will take longer than one-pass encoding, the improvement is well worth the time. In some situations, two-pass encoding can improve visual quality by as much as 30% when using VBR. If encoding time is a concern, a faster first pass can be achieved by keeping sub-pixel estimation complexity to a lower value such as 1 or 2, and the number of reference frames to 1.

The second pass should be slower by setting sub-pixel estimation complexity to 6 and the number of reference frames to 5.

Encoder buffer model

When using the x264 system output, bitrate can be controlled using VBV (Video Buffer Verifier). VBV is a theoretical MPEG video buffer model used to ensure that an encoded video stream can be correctly buffered and played back at the decoder device. By definition, the VBV should not overflow or underflow when its input is a compliant stream. When encoding such a stream, it is important that it complies with the VBV requirements. There are two parameters that need to be set for VBV controls — VBV buffer size and VBV maximum bitrate. Consider a small time period T over that desired to restrict the data throughput. We recommend a VBV buffer size of 1-5x video bitrate, and all renditions use the same VBV buffer size multiplier:

- VBV buffer duration = VBV buffer size / VBV maximum bitrate
- VBV data throughput = (T + VBV buffer duration) * VBV maximum bitrate

It is important to keep VBV data throughput well below the video maximum bitrate specification of the highest H.264 profile the device can support and the average net- work bandwidth available for video of end user.

Audio encoding

To ensure renditions in the multiple-bitrate set can be played back appropriately, you must encode each rendition's audio track using the same sample rate and bitrate. Using the same sample rate avoids issues such as audio/video sync, as most devices rely on the audio track’s time clock to control playback. Using the same bitrate avoids issues such as audible “pops” when the device changes from one rendition to another.

The audio codec used with H.264 is generally AAC.

Delivery best practices

Consider these recommendations for the following when delivering your video content to the client:

General

Standard MP4 files are recommended input source type, as opposed to standard MP4. In addition, the MP4 container is preferred over FLV.

Standard MP4 files have a single index of all keyframes that must be parsed before a stream can start. In most cases, FLV files do not have an index of keyframes, so seeking in the video is slower while the server searches to find the exact time to file offset mapping for each request.

Fragmented F4F files are also supported as a valid input source if the content is pre-encrypted using Adobe Access. Fragmented MP4 files have a nice balance because the index of keyframes is split and distributed throughout the fragments.

MP4 container

To support efficient streaming in Media Services On Demand, all MP4 containers should have their MOOV box placed at the beginning of the file, before any MDAT box but after any FTYP box. If the video file does have an MDAT box before the MOOV atom it will still play back, but you might experience a higher start-up time due to the round trip required to determine the exact location of the MOOV box.

Many encoders provide an option that ensures proper placement of this box, usually labeled with something like “Prepare for Streaming.” Content that does not follow this recommendation can be fixed using tools to move the MOOV atom to the proper location. Several free tools are available online that can correct this very quickly without the need to re-encode.

Fragmented MP4 container

When using Adobe System Incorporated HTTP Dynamic Streaming, fragment duration for all fragments should be between two and four seconds and should be constant across all fragments. The Adobe HTTP Dynamic Streaming specification requires fragments to begin with a keyframe. As a result, if the input F4F fragment duration is a larger value such as five or ten seconds, then the output fragment duration can be only a multiple of these values, resulting in very long output fragments. A smaller input fragment size allows more flexibility in configuring the output fragment size to an optimum duration value.

When creating a F4F package, ensure that you use the latest Adobe packager that has several fixed problems.

References

- *http://www.progettosinergia.com/flashvideo/finalMAX2011.pdf*
- *http://sonnati.wordpress.com/best-of-blog/*
- *http://mewiki.project357.com*

Encoding recommendations

In this section you'll find bitrate calculations and resolution recommendations.

Selecting bitrates

Your optimal bitrate will depend on your encoder and the features you have chosen to enable.

The following rough calculation provides you with a good baseline Kbps:

```
(FRAME_HEIGHT * FRAME_WIDTH * FRAME_RATE) / MOTION_FACTOR / 1024 = BASELINE
Kbps
```

where *MOTION_FACTOR* is:

- 7 for high-motion, high screen-change clips
- 15 for standard clips
- 20 for low-motion (talking head) clips

Note: This is just a suggested algorithm to get you started; your mileage might vary. Keep in mind that clips almost always tend to be higher motion than you originally anticipate. The outcome is also dependent on the features you enable.

Notice

Akamai secures and delivers digital experiences for the world's largest companies. Akamai's Intelligent Edge Platform surrounds everything, from the enterprise to the cloud, so customers and their businesses can be fast, smart, and secure. Top brands globally rely on Akamai to help them realize competitive advantage through agile solutions that extend the power of their multi-cloud architectures. Akamai keeps decisions, apps, and experiences closer to users than anyone — and attacks and threats far away. Akamai's portfolio of edge security, web and mobile performance, enterprise access, and video delivery solutions is supported by unmatched customer service, analytics, and 24/7/365 monitoring. To learn why the world's top brands trust Akamai, visit *www.akamai.com*, *blogs.akamai.com*, or *@Akamai* on Twitter. You can find our global contact information at *www.akamai.com/locations*.

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 57 offices around the world. Our services and renowned customer care are designed to enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers, and contact information for all locations are listed on *www.akamai.com/locations*.

Published 10/2020