Prefetching in AMD and ACE - Origin-Assist Interface

1. Background

Playback of streaming content is a very reactive process where the player requests an object â segment and manifest files â from the CDN and the CDN in turn fetches that object from the Origin. Once fetched from the Origin, the CDN will typically cache the object so that subsequent requests for the same object will not require going back all the way to the Origin.

From a CDN's point of view, not having the object in cache and hence, needing to go forward to the Origin, is called a cache-miss. On the contrary, if the object was indeed found in the CDN cache, it's called a cache-hit.

As the player continues playback and keeps fetching new objects from the CDN, this sequence of an initial cache-miss and then subsequent cache-hits repeats for all objects. Obviously, a cache-hit is a good thing as it eliminates the latency in fetching the object from the Origin.

But we can do better ?? If we can predict the next object that a player will request and **prefetch** it in the CDN's cache, every player will get a cache-hit for all, or most, of the objects.

The concept of **prefetching** strives to position objects at the "Edge of the Internet" in anticipation that these will be requested by the player imminently, thereby reducing the time to deliver that object to the player.

2. Introduction

Prefetching theoretically can be done in many ways and they can be generally categorized into three:

Origin-Assist

Whenever CDN goes forward to the Origin to fetch an object, the response additionally includes a new response header that indicates to the CDN the next object that will be requested by the player and hence, can trigger the prefetch of that anticipated object.

Basically CDN relies on some assistance from an intelligent Origin to trigger prefetching.

Smart CDNÂ

In this scheme, given the current request and possibly some other external/historical state/input, CDN is able to figure out the URL of the "next" object and hence, trigger prefetching.

Player-Assist

This is similar to theÂ *origin-assist*Â scheme, except the player provides information of the "next" object in the request for the "current" object.

This document describes the HTTP based interface for the **Origin-Assist** approach that can be used by an Origin to trigger prefetch of objects within Akamai's AMD (Adaptive Media Delivery) and ACE (Akamai Cloud Embed) products.

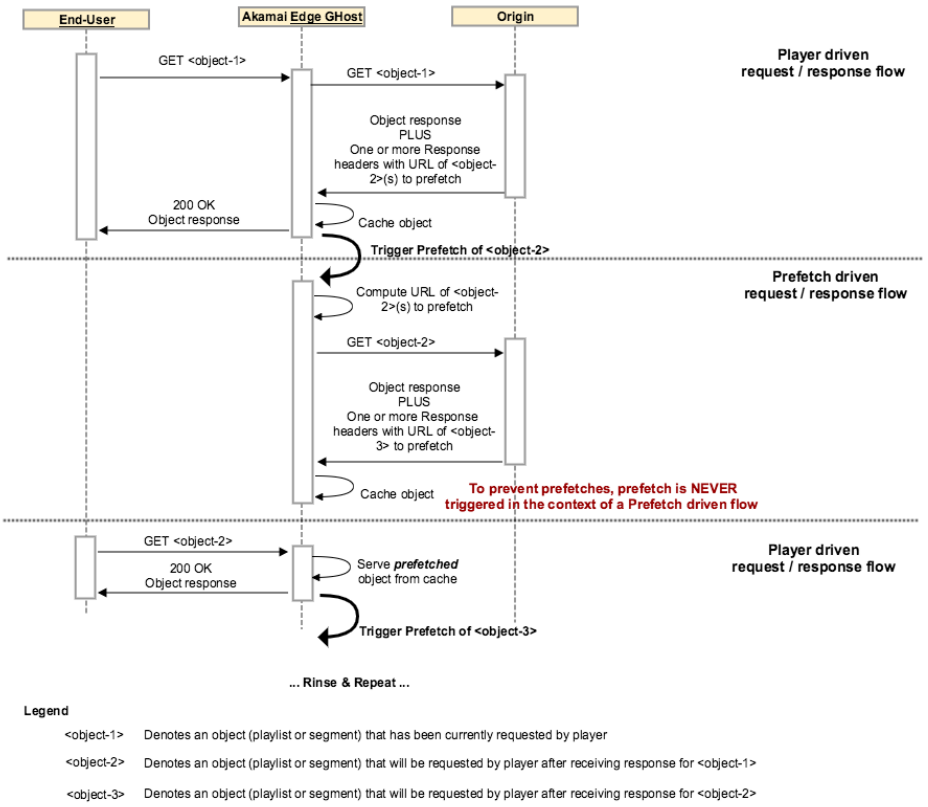
> While there are references to Akamai's AMD/ACE streaming product, the spec is generic and can be implemented by any Origin against any CDN.

3. Prefetch Interface

3.1. High Level Flow

The following sequence diagram shows the request/response flows when a player/end-user makes a request for an object to AMD/ACE and how the prefetch behavior is triggered off of that.

1. **Player makes a request for <object-1>**
2. AMD/ACE handles that request by forwarding the request to Origin.
3. Origin returns the requested object, <object-1>, back to AMD/ACE.
    a. Origin response also includes headers to prefetch <object-2>. The value of this header is either an absolute URL path OR a relative URL path (relative to <object-1>)
4. AMD/ACE returns <object-1> to player and thus completes the request/response flow for <object-1>
5. Since Origin's response for <object-1> included prefetch response headers for <object-2>, AMD/ACE will trigger prefetching of <object-2>
    a. AMD/ACE creates a HTTP/S request for <object-2> and forwards it to Origin. This request is called a 'prefetch request'
    b. This prefetch request from AMD/ACE will also contain a pre-configured request header, informing Origin that this is a prefetch request.
6. Origin returns the requested object, <object-2>, back to AMD/ACE
    a. Origin response could include response header to prefetch <object-3>
7. AMD/ACE simply caches <object-2>
    a. AMD/ACE will NOT trigger prefetching for <object-3>, since, <object-2> was retrieved from Origin in the context of a prefetch request (else we will end up in a never ending recursive prefetching)
8. **Player makes a request for <object-2>**
9. AMD/ACE handles the request by fetching <object-2> from it's cache and returns the response back to player
    a. Since, response for <object-2> from Origin contained prefetch response headers for <object-3>, AMD/ACE will now trigger prefetching of <object-3>



3.2. Interface Between Origin and AMD/ACE

3.2.1. Request Header from AMD/ACE to Origin to indicate support for Prefetching

When AMD/ACE makes a request to the Origin, a request header indicating that prefetching is enabled in AMD /ACE will be sent to Origin. Origin could use this as a trigger to configure a response header for the 'next' object to prefetch and send it back to AMD/ACE.

This request header sent by AMD/ACE to Origin is:

```
CDN-Origin-Assist-Prefetch-Enabled: 1
```

3.2.2. Response Header from Origin to Communicate Object to Prefetch

For an origin to trigger prefetching in AMD/ACE, the origin shall return one or more response headers, identically named, as shown below:Â

```
CDN-Origin-Assist-Prefetch-Path: <relative|absolute path of the object's URL that
should be prefetched>
```

Upon receipt of the response from Origin, including the above mentioned prefetch response header, AMD/ACE will trigger prefetching of the object available at the URL mentioned in the response header.

IMP:Â The URL of the object that will be prefetched by AMD/ACE will be created using:

a) the player-requested URL of the current object and
b) the relative or absolute path listed in the prefetch response header.

3.2.2.1. Â Single Prefetch-able Object URL Per Response Header

The Origin server can return a single response header with URL for the a single object to be prefetched.

If multiple objects are to be prefetched, then one response header per prefetch-able URL can be returned by Origin. All response headers are named the same. The order of these response headers will control the order in which the requests for the prefetch-able URLs are issued by AMD/ACE.

> ⚠ The caveat is that the ordering of these response headers may not be preserved by middle layer proxies.

Example

Request from AMD/ACE to Origin

```
GET <object-url-path> HTTP/1.1
```

Response from Origin to AMD/ACE

```
HTTP/1.1 200 OK
...blah...
CDN-Origin-Assist-Prefetch-Path: <relative|absolute path of prefetch-able object's URL>
CDN-Origin-Assist-Prefetch-Path: <relative|absolute path of prefetch-able object's URL>
Content-Length: x
...blah...


<object body>
```

3.2.2.2. Multiple Prefetch-able Object URLs Per Response Header

As an alternative to 'Single Prefetch-able Object URL Per Response Header', the Origin server can return a single response header with one or more URLs, comma delimited, for one or more prefetch-able objects.

Example

Request from AMD/ACE to Origin

```
GET <object-url-path> HTTP/1.1
```

Response from Origin to AMD/ACE

```
HTTP/1.1 200 OK
...blah...

CDN-Origin-Assist-Prefetch-Path: <relative|absolute path of prefetch-able object's
URL>,<relative|absolute path of prefetch-able object's URL>

Content-Length: x

...blah...


<object body>
```

IMP NOTE: The comma character is treated as a delimiter, as such, if one or more comma characters are present in the URL path to be prefetched, they should be escaped as %2C by url encoding

3.2.3. Request Header from AMD/ACE to Origin to Prefetch Object

Upon receiving a prefetch response header from Origin, assuming that prefetching is enabled, AMD/ACE will request that object from Origin as if the object was requested by player.

To allow for Origin to distinguish between a prefetch request VS a regular request, all prefetch requests will have the following request header:

```
CDN-Origin-Assist-Prefetch-Request: 1
```

3.2.4. URL Construction in AMD/ACE for Triggering Prefetch

The following sub-section is INFORMATIONAL and shows how the relative and absolute paths in the prefetch response headers are used in AMD/ACE to construct a prefetch request.

3.2.4.1. Absolute Path

An absolute path can be present in the response header and will be indicated via a **leading forward slash**. In this case, the complete string shall be used as is.

Example:

URL of Player Initiated Request	Value of Response Header from Origin (meant to trigger prefetch)	URL of Request Prefetched from AMD /ACE
https://property-hostname /some/1234/video-100k-pl. m3u8	/hls/live/1234/video-100k/seg1.ts	https://property-hostname-from-player-request/**hls/live/1234/video-100k/seg1. ts**
https://property-hostname/thi ng/1234/video-100k/seg1.ts	/hls/live/1234/video-100k/seg2.ts	https://property-hostname-from-player-request**/hls/live/1234/video-100k/seg2. ts**

where:

property-hostnameÂ is the HTTP/1.1 'Host' header as seen by AMD/ACE

property-hostname-player-request is the same as above

3.2.4.2. Relative Path

A relative path can be specified in the response header and will be indicated via the **absence of a leading forward slash**. In this case, the forward path of current request, minus the filename, will be used as the base path for construction of the prefetch URL path.

Example:

URL of Player Initiated Request	Value of Response Header from Origin (meant to trigger prefetch)	URL of Request Prefetched from Akamai GHost
https://property-hostname/some/1234/video-100k-pl.m3u8	video-100k/seg1.ts	https://property-hostname/some/1234/**video-100k/seg1.ts**
https://property-hostname/thing/1234/video-100k/seg1.ts	seg2.ts	https://property-hostname/some/1234/video-100k/**seg2.ts**

where:

property-hostnameÂ is the HTTP/1.1 'Host' header as seen by AMD/ACE

property-hostname-player-request is the same as above

3.3. Examples

3.3.1.1. Trigger Prefetch of Video-only and Audio-only Playlists on request for HLS master manifest

```
GET /hls/live-streaming/fifa/france-croatia/master.m3u8 HTTP/1.1
CDN-Origin-Assist-Prefetch-Enabled: 1


HTTP/1.1 200 OK
...blah..
CDN-Origin-Assist-Prefetch-Path: /hls/live-streaming/fifa/france-croatia/video-1000k
/pl.m3u8
CDN-Origin-Assist-Prefetch-Path: /hls/live-streaming/fifa/france-croatia/audio/pl.
m3u8
...blah...


<object body>
```

Call-outs

- Request header indicating that prefetch functionality is enabled in AMD/ACE
- 2 prefetch specific response headers are present in the response.
- Value of both response headers is a string that starts with a forward slash, i.e. this is an example of an absolute URL
- The above response header will trigger following prefetch requests on the same host as that of the HLS master manifest

```
GET /hls/live-streaming/fifa/france-croatia/video-1000k/pl.m3u8 HTTP/1.1

GET /hls/live-streaming/fifa/france-croatia/audio/pl.m3u8 HTTP/1.1
```

3.3.1.2. Trigger Prefetch of Video Segment on request for Video Playlist

```
GET /hls/live-streaming/fifa/france-croatia/video-1000k/pl.m3u8 HTTP/1.1

CDN-Origin-Assist-Prefetch-Enabled: 1

CDN-Origin-Assist-Prefetch-Request: 1


HTTP/1.1 200 OK

...blah..

CDN-Origin-Assist-Prefetch-Path: seg1.ts

...blah...


<object body>
```

Call-outs

- Request header indicating that prefetch functionality is enabled in AMD/ACE
- Request header indicating that this is a prefetch request
- 1 prefetch specific response header is present in the response.
- Value of this response header is a string that doesÂ **not** start with a forward slash, i.e. this is an example of an relative URL
- The above response header will trigger following prefetch requests on the same host as that of the video playlist

```
GET /hls/live-streaming/fifa/france-croatia/video-1000k/seg1.ts HTTP/1.1
```