Blink LazyFrames

sclittle@chromium.org
Last modified 2018-01-29

The LazyFrames mechanism in Blink defers certain iframes from being loaded until the user scrolls near them, in order to save data, speed up the loading of other parts of the page, and reduce memory usage. It will be used as part of the LazyLoad feature.

Detailed design

Which iframes should be deferred?

The goal is to defer iframes that are intended to be shown to the user (e.g. banner ads) until the user scrolls near them, but leave iframes used for communication (e.g. for social media widgets) or analytics intact, since deferring communications/analytics frames would break that functionality.

An iframe will be deferred if it satisfies all of the following:

- It's a third-party iframe (i.e. a different origin than the embedding page),
- Larger than 4x4 in dimensions,
- Not marked as "display:none" nor "visibility:hidden",
- Not positioned off-screen using negative x or y coordinates

First-party iframes share a javascript context with the embedding page, so deferring them could break the page. The other heuristics are used to detect if the iframe would be visible to the user, since iframes used for analytics or communication typically are a combination of tiny, invisible, and/or positioned off-page.

When an iframe appears on a page, a histogram Blink.LazyFrames.DeferralAction will have different enum values recorded according to which of the above reasons caused the iframe to not be deferred (plus an additional reason for when the iframe is already near enough to the viewport to get loaded) or record that the iframe was indeed deferred.

Deferring the frame

In HTMLFrameOwnerElement::LoadOrRedirectSubframe(), before calling FrameLoader::Load() on the child frame, check the above conditions. If the frame should not be deferred or is already visible, call FrameLoader::Load() like normal, otherwise install an IntersectionObserver on the HTMLFrameOwnerElement that will trigger once the user scrolls near the element. The actual distance threshold used will be specified by the overall LazyLoad feature, and determined through experimentation.

Resuming the frame

If the IntersectionObserver on the HTMLFrameOwnerElement triggers, then call a new method HTMLFrameOwnerElement::LoadFrameWhenVisible(), which will call FrameLoader::Load() on the child frame.

Recording data savings

Data savings will initially be recorded as a coarsely estimated 50KB per deferred frame, sending a message via mojo to the data reduction proxy component in the browser process each time a frame is deferred. For each deferred frame that gets resumed, a message will be sent via mojo to the data reduction proxy which will cancel out the previously reported savings. 50KB is a rough and conservative estimate determined anecdotally by manually browsing sites and adding up the content lengths of subresources inside a few banner-ad iframes, and then subtracting some bytes to account for browser cache hits.

In future versions, the actual data usage of these kinds of iframes will be measured, and the estimated savings per deferred frame will be determined based on that. Chrome doesn't measure exact actual data savings since that would require loading the iframe in order to get the full set of subresources inside it.