First (Non-Blank) Paint & First Contentful Paint

First indications of the page load

[Public]
Created: July 21, 2016
Owner: Paul Irish
Status: draft

Background

Page load is not about any one particular timestamp, but a number of timings and their relationship paint a picture of the journey of loading a page. Two of the earliest indications of a loaded page are the First Paint and First Contentful Paint. These complement other metrics like Time to First Meaningful Paint and Time to Interactive.

First (Non-Blank) Paint

When anything beyond the document background is painted.

Has low value to a user, as this paint may be completely meaningless or useless. The first non-blank paint is often represented as "Start Render" in performance products.

Definition: The timestamp of a shipped frame that contains the first paint to draw more than the document background.

First Contentful Paint

First Contentful Paint is the time when some contentful thing (text, image, canvas, or SVG) is painted for the first time. It can often be non-meaningful paints like headers and navigation bars.

Definition: The timestamp of a shipped frame that contains ANY of these:

- First paint of text
- First paint of SVG
- First paint of an image
- First paint of a canvas

Implementation in Chromium

(This section borrows heavily from ksakamoto's Time-to-first-X-paint metrics: status and refinement plans)

The Time-to-first-X-paint metrics are instrumented in Blink graphics layer (platform/graphics). The following diagram illustrates how time-to-first-text-paint timing is instrumented and reported. (The other first-X-paint metrics follow similar pattern.)

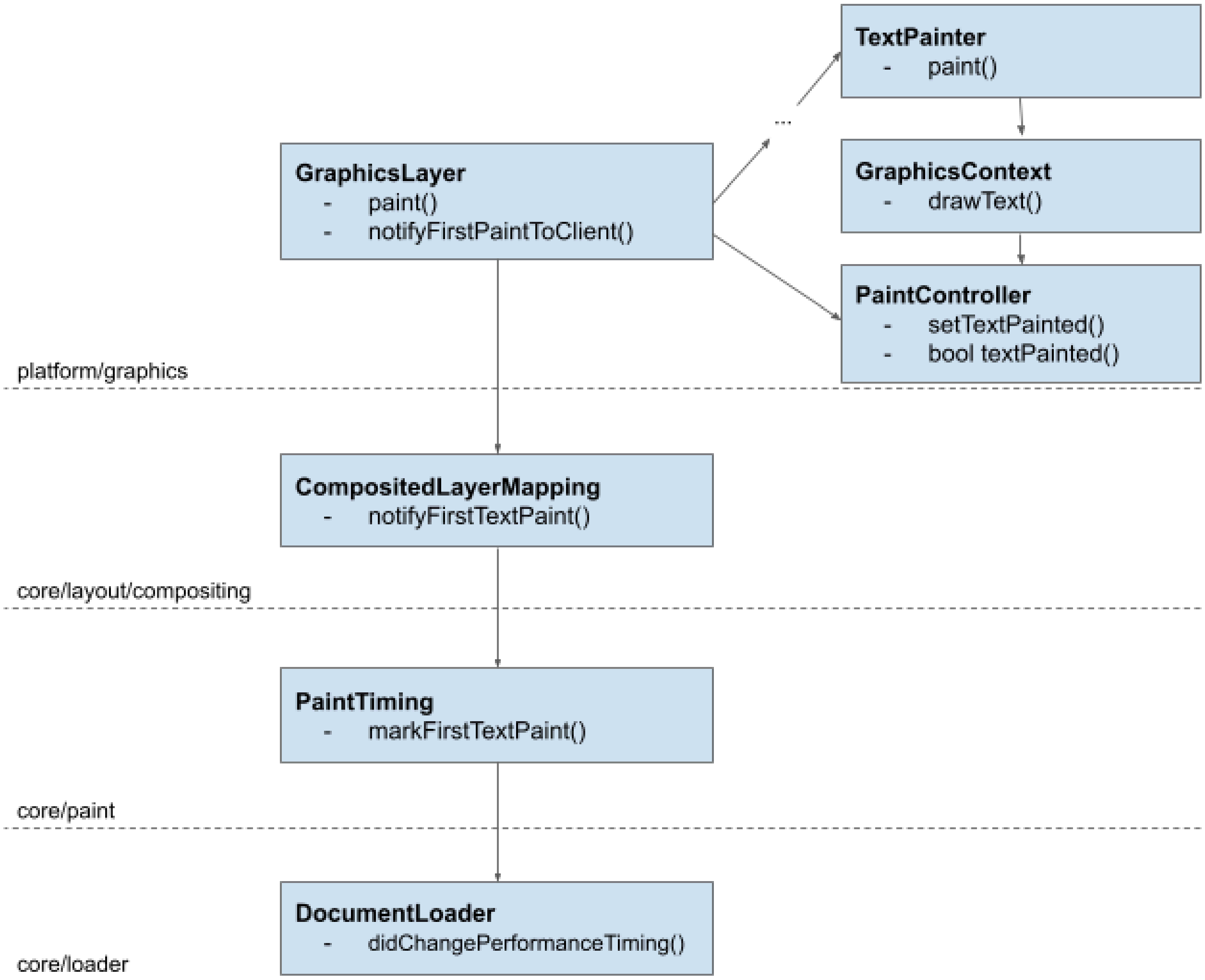

***Figure 1**: Reporting path of Time-to-First-Text-Paint*

GraphicsLayer::paint() is the entry point of painting operation for a layer. Inside this function, paint() function of each painter object is called with a GraphicsContext argument. Painter object calls drawing functions on GraphicsContext, such as drawText(). If drawText() actually painted any text, GraphicsContext calls PaintController::setTextPainted() to indicate that this layer has painted text.

Before returning from GraphicsLayer::paint(), GraphicsLayer::notifyFirstPaintToClient() is called. It checks the flag on PaintController, and calls CompositedLayerMapping:: notifyFirstTextPaint() if text is painted on the layer for the first time. The notification is forwarded to PaintTiming, which is a Document supplement. PaintTiming notifies DocumentLoader of per-document first text paint timing. Here, it converges with other document-related timings from DocumentTiming, and reported to the PageLoadMetrics via WebPerformance.

FP Implementation:

UMA:		PageLoad.PaintTiming.NavigationToFirstPaint
Trace event:	blink.user_timing.firstPaint
firstPaint in PaintTiming.h

This is recorded when GraphicsLayer::paint() is called for the first time.
PaintTiming::markFirstPaint()

FCP Implementation

UMA:		PageLoad.PaintTiming.NavigationToFirstContentfulPaint
Trace event:	blink.user_timing.firstContentfulPaint
firstContentfulPaint in PaintTiming.h

Minimum of firstTextPaint, firstImagePaint, first non-empty canvas, first SVG paint

- firstTextPaint: Triggered by GraphicsContext::drawText() and drawBidiText(). This does not include invisible text for downloading web fonts.
- firstImagePaint: Triggered by GraphicsContext::drawImage(). For images that render progressively, this is triggered as soon as any pixels have been drawn.

PaintTiming::markFirstContentfulPaint()

Evaluation

Time-to-first-X-paint metrics discusses an evaluation comparing these metrics against human reaction to a 200 page loads via WebPageTest. It identifies issues around blank paints and hidden images.

Timing Disparity Note

The existing Chromium implementations for these timings mark the time at which these are painted on the main thread (inside of `GraphicsLayer::paintWithoutCommit`). The actual frame is delivered at SwapBuffers a few ms later. Early in the page load, it's likely the delta between Paint and SwapBuffers is < 20ms, however with CPU contention and large rasterization costs, this could grow to a disparity of > 400ms.

DevTools Disparity Note

Chrome DevTools reports a first paint in the Timeline. This paint is currently not observed from the firstPaint trace event, but rather synthesized from the *first Composite Layers event after CommitLoad* fired. (source: TimelineModel.js)

See Also

First Paint as a WebPerf API

- Why First Paint as Web Perf API?
- Pl include firstPaint time in navigation timings · Issue #45 · w3c/navigation-timing

Related efforts

- Progressive Web Metrics: Living Spec
- tdresser/time-to-first-meaningful-paint explainer
- tdresser/time-to-interactive explainer
- Hero Element Timing API
- Progressive Web Metrics - Google Groups
- Long Tasks V2 API