Skip Forced Style Update on Parsing Finished

Attention: Shared Google-externally

Author: futhark@chromium.org

Last Updated: 2019/05/24

TL;DR An attempt to get rid of the forced `Document::UpdateStyleAndLayoutTree` call from `Document::FinishedParsing` to fix issues like 332189 and 742413. This document proposes a way to do this by making sure we call `Document::UpdateStyleAndLayoutTree` at least once after we finish parsing the document before we trigger the load event.

Current state

The problem is two-fold. There is a performance side and a correctness side.

If we update style and layout tree while we load render blocking resources, we will still have to recalculate style and rebuild the layout tree when the resources finishes loading.

We do not check if there is anything blocking rendering (like stylesheets) when calling UpdateStyleAndLayoutTree from FinishedParsing. If we block rendering on a stylesheet which contains transitions, we may end up transitioning between UA style and the author applied style after the stylesheet finishes loading (see wpt test).

Also, we currently have inconsistent load event blocking since this case would not block onload for frame.html:

```
<link rel=stylesheet href=object-is-block.css>
<style>object { display: none }</style>
<object data=frame.html />
```

while this would:

```
<style>object { display: block }</style>
<link rel=stylesheet href=object-is-none.css>
<object data=frame.html />
```

Note that Firefox loads resources for display:none object elements and blocks onload for them.

Proposed solution

We cannot simply remove the forced update without making sure we are delaying the load event as specified in the html spec. Instead, keep track of whether we have done an UpdateStyleAndLayoutTree after parsing finished and force one just before checking if we should trigger the load event to check if any load event blocking resources start loading.

~~There are cases where this might fire load events later than the current behavior if no render blocking resources are loading when parsing finishes. It's possible that we should still call UpdateStyleAndLayoutTree on parsing finished if IsRenderingReady() is true.~~

Alternative solution(s)

An alternative solution would be to follow the Firefox behavior which loads the <object> resources regardless of whether they are rendered or not. That would be a pretty big change to how we load those resources, and it does not seem to not match the spec for <object>. It is also possible that we could do a more selective check for elements which triggers load event blocking resource loading, but that sound complex and error prone.

Work so far

There have been a few cases where production code or tests have been relying on the style and layout tree being up-to-date at the point when parsing the html is finished. All such issues have been fixed:

https://chromium-review.googlesource.com/c/chromium/src/+/1624330

https://chromium-review.googlesource.com/c/chromium/src/+/1624565

https://chromium-review.googlesource.com/c/chromium/src/+/1627355

https://chromium-review.googlesource.com/c/chromium/src/+/1624193

Implementation of the proposed solution

We have an implementation behind a flag which skips the forced update if that flag is enabled, flips a flag in Document when UpdateStyleAndLayoutTree is called after parsing is finished, and if we still haven't done the update when checking if we can trigger the load event, do an UpdateStyleAndLayoutTree before dispatching the event:

https://chromium-review.googlesource.com/c/chromium/src/+/1626589

Possible next steps

- Land the implementation behind a flag
- Make wpt cases for onload behavior and see what interop looks like
- Run an experiment with the flag enabled