Improved load-time scheduling

Feb 16th, 2017
skyostil@

Tracking bug: TODO(skyostil@)

This is a proposal for a more optimal load time scheduling policy in the renderer. By using the recently introduced loading use case detection in the renderer scheduler we hope to make page loading more efficient.

Possible policies to try:

- Figure out if we are detecting the loading use case correctly in the first place.
- Make loading tasks high priority.
- Deprioritize loading offscreen iframes.
- Make compositing tasks low priority.
- Make compositing tasks best effort.
- Limit rendering frame rate.
- Prioritize timers vs. loading tasks differently.
- Use resource priorities for loading tasks.
- (Detect idle foreground tabs for v8.)

Evaluation techniques:

- Manual testing and tracing on popular sites.
- Side-by-side testing on two identical phones -- does it “feel” like the page loads faster when we schedule tasks differently?
- Run page_cycler_v2.typical_25 and see if any of this affects page loading time.
- Check with smoothness.top_25_smooth if there’s an impact on responsiveness.

Risks:

- Loading use case detection is based on the types of layouts we do (i.e., rendering/compositing tasks). If we deprioritize them too much, we may never get out of the loading state.