Incremental property tree updates

Philip, Xianzhu
October 2016

Blink's paint property trees should be built & updated as-needed instead of fully reconstructing them on every frame. This is a necessary step towards shipping paint properties which are used for geometry mapper, slimming paint v2, and slimming paint invalidation.

Design

A dirty bit pattern will be used to mark objects as needing their paint properties updated. Properties will need an update for any change that affects them directly (e.g., transform values change) as well as changes to ancestor properties referenced by them (e.g., transform parent goes away) [1]. Paint properties will still get updated during the PrePaintTreeWalk (shared by property tree updating and paint invalidation) but entire subtrees can be skipped if no paint invalidation or property updates are needed.

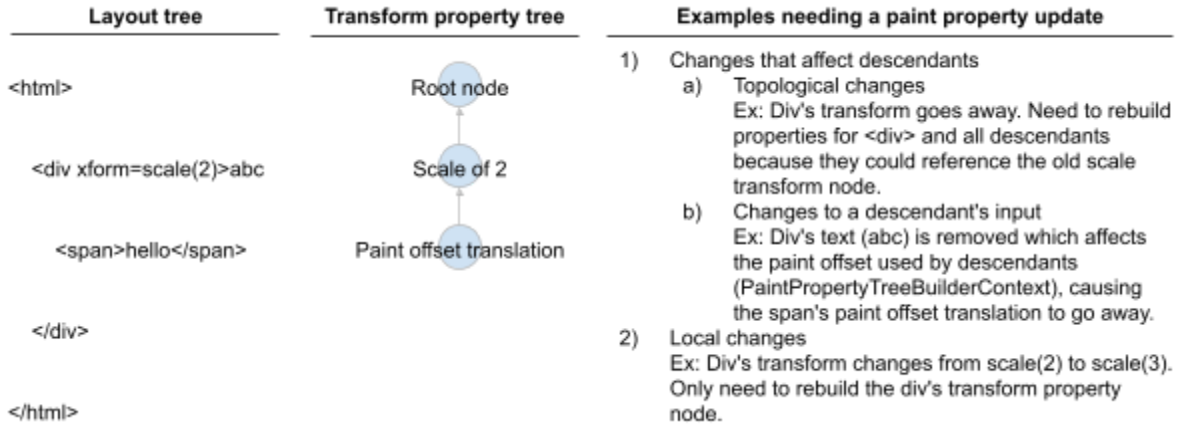
Tracking bug: https://bugs.chromium.org/p/chromium/issues/detail?id=645667
Proof of concept: https://codereview.chromium.org/2404213004

Implementation plan

1) Add performance testing to track improvements to the overall PrePaintTreeWalk performance.
   https://crrev.com/e500e8ba8aac6f63a94ba7a411b536f513d2270d
2) Refactor PaintPropertyTreeBuilder's update functions to use a pattern where it is easy to skip property update work.
   https://crrev.com/95247e236ea9fcb826b8f50e3cecdb2082833aee
3) Start tracking whether an object or descendants need property updates and skip property updates where possible. This flag always rebuilds entire property subtrees, but will be improved later. This will also include asserts to catch cases where an object needed property updates but wasn't marked as such, similar to paint under-invalidation tracking.
   https://crrev.com/1f505949fff2006a80f3883f129c580cf423c67c
4) Improve the "needs update" tracking by supporting local-only property updates that do not invalidate the entire property subtree. This tracking will need to consider paint offset changes as well as property tree additions/removals.
   https://crrev.com/2459701934af4c8297ee26d351a1fbf9596c6bd5
5) Stop the prepaint tree walk in places where no subtree paint invalidation or subtree property updates are needed.
   https://crrev.com/96606ec518ebf57313b7a9cb692905da3ee76b27

Appendix

[1] A simplified example below shows the kinds of property tree changes this proposal handles:



Objects will be marked as needing a property update in many places:

1) When paint offset or location changes. We re-use the paint invalidation system for this.
2) If there is a topological change to an ancestor in the property tree.
3) During update style when transform/opacity/effect/etc change.
4) When scroll offsets and scrolling contents size change.
5) Probably a few others...

[2] Miscellaneous thoughts

1) We need to move the property trees (ObjectPaintProperties.h) onto LayoutObject for performance reasons, as they are currently stored in a static hashmap to avoid bloating LayoutObject.
2) Which will be more valuable: skipping property tree updates (but continuing the context-updating walk), or fully pruning the prepaint tree walk when possible?
3) How granular should needsPaintPropertyUpdate flags be? We could specify that an object needs only the local effect node recomputed, for example.