BlinkGenPropertyTrees

Chris Harrelson (chrishtr), David Bokan (bokan), Jianpeng Chao (chaopeng), Mason Freed (masonfreed), Majid Valipour (majidvp), Peter Mayo (petermayo), Philip Rogers (pdr), Robert Flack (flackr), Sahel Sharifymoghaddam (sahel), Stephen McGruer (smcgruer), Tien-Ren Chen (trchen), Vladimir Levin (vmpstr), Xianda Sun (sunxd), Xianzhu Wang (wangxianzhu), Xida Chen (xidachen)

Blink-generated property trees (BGPT) is a project to simplify graphical compositing data structures in preparation for making compositing decisions after paint. BGPT updates the interface between Blink and the compositor from a complex layer tree to property trees + a drawable layer list [1]. It is launched in M75 (est: beta May 2, 2019, stable Jun 4, 2019). Launch bug: https://crbug.com/836884.

Goals

- Separate cc::Layer hierarchy information from drawable content. This is required for the paint team's primary project of making compositing decisions after paint because cc::Layer creation will be moving post-paint while hierarchy information (i.e., property trees) is determined pre-paint.
- Instead of sending a cc::Layer tree from blink to cc, only send a drawable cc::Layer list. With this change, many fields on cc::Layer can be removed.
- Unify property tree construction. Property trees were built twice, once in blink and again in cc. With BGPT, property trees are no longer built by cc and are copied (with a conversion step) from blink to cc by PaintArtifactCompositor.
- Unblock projects that de-duplicate composited and non-composited logic. One example is scroll unification. Another example is paint-changing compositor effect such as raster-inducing scroll.

Implementation

The major changes are:

- In pre-paint, Blink now creates *complete* transform, clip, effect, and scroll property trees. This includes property tree nodes that are not used in blink such as the viewport page scale transform node.
- During paint, Blink has an additional step to collect all painted GraphicsLayers as a list of ForeignLayers. This is an incremental staging towards CompositeAfterPaint (CAP) which uses compositing decisions from the existing compositor (PaintLayerCompositor which produces GraphicsLayers) with the new CAP compositor (PaintArtifactCompositor). While the new CAP compositor can make compositing decisions, this is overridden by forcing compositing for ForeignLayers.
- After paint, Blink copies the list of ForeignLayers to cc using PaintArtifactCompositor.
- After paint, Blink copies property trees to cc using PaintArtifactCompositor.

- Document animations now take a list of compositing decisions as input. With this change, animations updates have been moved post-paint.
- Cc no longer builds property trees in cc/trees/property_tree_builder.cc.

Results

Overall, BlinkGenPropertyTrees was flat on UMA performance. There was a significant regression in render surfaces due to https://crbug.com/947715 which was addressed with a border radius shader (FastBorderRadius). For more information about performance, see: BlinkGenPropertyTrees performance analysis.

Simpler code: **13,428 lines of code** have been removed following the launch.

Many bugs were fixed that were difficult or expensive to fix before:

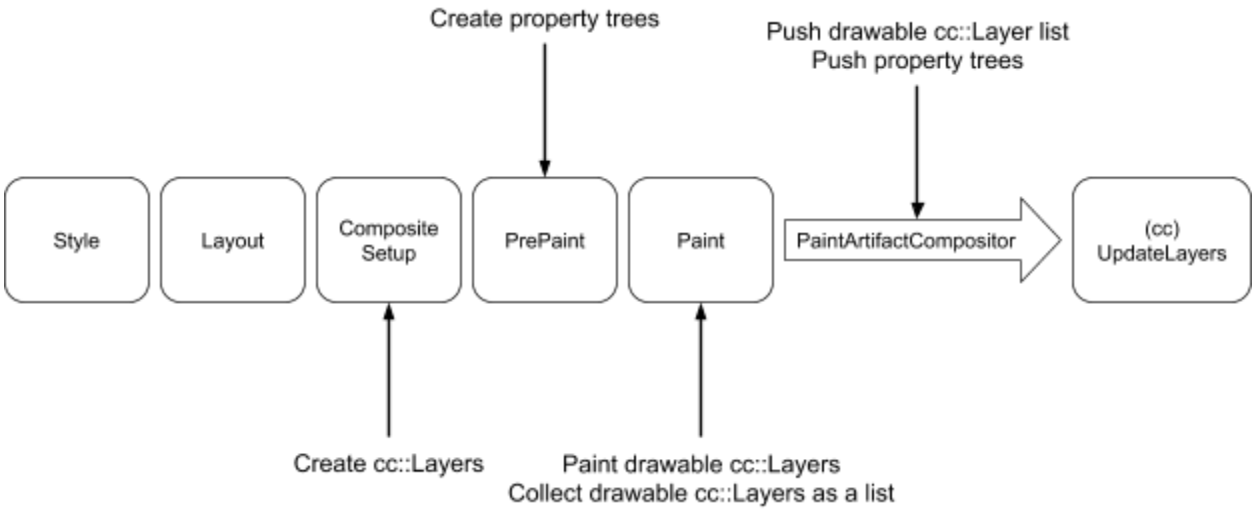
crbug.com/901083 - Outer viewport clip node has incorrect bounds
crbug.com/678669 - Out-of-flow positioned elements cannot escape composited border-radius mask
crbug.com/897559 - page render (layout) is incorrect when some elements is fixed.
crbug.com/467484 - Composited content under <foreignObject> does not position correctly. (26 stars!)
crbug.com/944330 - Elements inside SVG foreignObject are not positioned correctly without BGPT (Hotlist-Google)
crbug.com/919359 - Composited position:fixed in composited scroller is clipped incorrectly
crbug.com/865039 - compositing modifies mask and filter
crbug.com/400829 - Video is not sized correctly by object-fit (19 stars!)
crbug.com/907175 - Filtering should apply before css clip and css clip-path
crbug.com/792280 - Composited clip-path applies in wrong order.
crbug.com/947570 - Content escapes border-radius clip with composited animations

Addendum

Related documents:

- BlinkGenPropertyTrees launch retrospective (June 12, 2019)
- BlinkGenPropertyTrees performance analysis
- Simming paint summary
- core paint/README.md and platform paint/README.md
- Rendering-core BlinkGenPropertyTrees update (Original target: M70!) Aug, 2018
- Summary of Composite-After-Paint architecture (Part of a BlinkOn 9 talk, April, 2018)
- Companion talks about property trees by ajuma and pdr (early 2017):
    - Compositor Property Trees
    - Blink Property Trees

[1] The major main-thread steps with BlinkGenPropertyTrees:



This change fits into the overall blink & compositor pipeline in the following way: